Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence (AAAI-10)

Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners

Miquel Ramírez
Universitat Pompeu Fabra
08018 Barcelona, SPAIN
miquel.ramirez@upf.edu

Hector Geffner
ICREA & Universitat Pompeu Fabra
08018 Barcelona, SPAIN
hector.geffner@upf.edu

Abstract

Plan recognition is the problem of inferring the goals and plans of an agent after observing its behavior. Recently, it has been shown that this problem can be solved efficiently, without the need of a plan library, using slightly modified planning algorithms. In this work, we extend this approach to the more general problem of *probabilistic* plan recognition where a probability distribution over the set of goals is sought under the assumptions that actions have deterministic effects and both agent and observer have complete information about the initial state. We show that this problem can be solved efficiently using classical planners provided that the probability of a partially observed execution given a goal is defined in terms of the cost difference of achieving the goal under two conditions: complying with the observations, and not complying with them. This cost, and hence the posterior goal probabilities, are computed by means of two calls to a classical planner that no longer has to be modified in any way. A number of examples is considered to illustrate the quality, flexibility, and scalability of the approach.

Introduction

The need to recognize the goals and plans of an agent from observations of his behavior arises in a number of tasks including natural language, multi-agent systems, and assisted cognition (Cohen, Perrault, and Allen 1981; Pentney et al. 2006; Yang 2009). Plan recognition is like planning in reverse: while in planning the goal is given and a plan is sought; in plan recognition, part of a plan is given, and the goal and complete plan are sought.

The plan recognition problem has been addressed using parsing algorithms (Geib and Goldman 2009), Bayesian network inference algorithms (Bui 2003), and specialized procedures (Kautz and Allen 1986; Lesh and Etzioni 1995; Huber, Durfee, and Wellman 1994). In almost all cases, the space of possible plans or activities to be recognized is assumed to be given by a suitable library or set of policies. Recently, an approach that does not require the use of a plan library has been introduced for the 'classical' plan recognition problem where actions are deterministic and complete information about the initial state is assumed for both the agent and the observer (Ramirez and Geffner 2009). In this approach, a goal G is taken to account for the observations when there is an optimal plan for G that satisfies them. The goals that account for the observations are selected from a set of possible goals, and are computed using slightly modified planning algorithms.

The advantages of this 'generative approach' to plan recognition, are mainly two. First, by not requiring a library of plans but a domain theory from which plans are constructed, the approach is more flexible and general. Indeed, (acyclic) libraries can be compiled into domain theories but not the other way around (Lekavý and Návrat 2007), and joint plans, that pose a problem to library-based approaches are handled naturally. Second, by building on state-of-the-art planning algorithms the approach scales up well, handling domains with hundred of actions and fluents quite efficiently.

An important limitation of our earlier account (Ramirez and Geffner 2009), however, is the assumption that agents are 'perfectly rational' and thus pursue their goals in an optimal way only. The result is that goals G that admit no optimal plans compatible with the observations are excluded, and hence, 'noise' in the behavior of agents is not tolerated.

The goal of this work is to introduce a more general formulation that retains the benefits of the generative approach to plan recognition while producing *posterior probabilities* $P(G|O)$ rather than boolean judgements. For this, a prior distribution $P(G)$ over the goals G is assumed to be given, and the likelihoods $P(O|G)$ of the observation O given the goals G are defined in terms of *cost differences* computed by a classical planner. Moreover, in contrast to our earlier formulation, the classical planner can be used off-the-shelf and does not need to be modified in any way.

The paper is organized as follows. We first go over an example and revisit the basic notions of planning and plan recognition from the perspective of planning. We then introduce the probabilistic formulation, present the experimental results, and discuss related and future work.

Example: Noisy Walk

Figure 1 illustrates a plan recognition problem over a grid 11x11 where an agent, initially at the center of the bottom row (cell marked I) heads to one of the possible goals A, B, C, D, E, or F, by performing two types of moves: horizontal and vertically moves at cost 1, and diagonal moves at

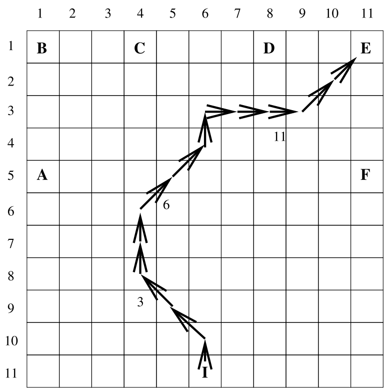

Figure 1: Noisy Walk: Observed path and possible goals

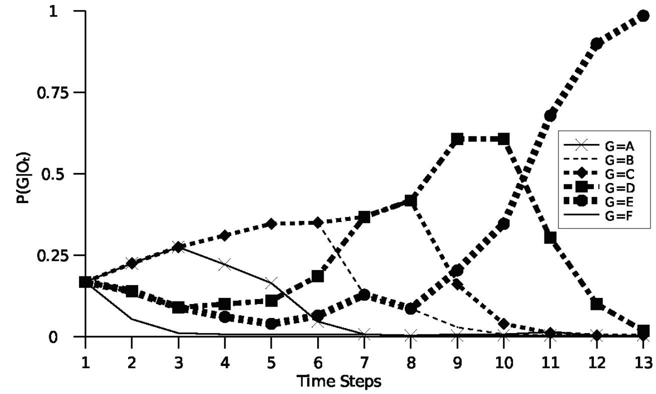

Figure 2: Noisy Walk: $P(G|O_t)$ as function of time t

cost $\sqrt{2}$. The arrows show the path taken by the agent, with the numbers 3, 6, 11 indicating the times at which some of the actions were done. As it can be seen, while the agent is pursuing and eventually achieves E, it's not taking a shortest path. Moreover, since the observed execution is not compatible with any optimal plan for any goal, the formulation in (Ramirez and Geffner 2009) would rule out all the hypotheses (goals).

The formulation developed in this paper does not prune the possible goals G but ranks them according to a probability distribution $P(G|O)$ where O is the observed action sequence. This distribution as a function of time is shown in Figure 2. As it can be seen, until step 3, the most likely goals are A, B, and C, after step 7, D and C, and after step 11, just E. The probabilities $P(G|O_t)$ shown in the figure for the various goals G and times t result from assuming uniform priors $P(G)$ and likelihoods $P(O|G)$ that are a function of a *cost difference*: the cost of achieving G while complying with O, and the cost of achieving G while not complying with O. For example, right after step $t = 6$, the cost of achieving A while complying with O is $7 + 3\sqrt{2}$, while the cost of achieving A while not complying with $\overline{O}$ is $1+5\sqrt{2}$. From this cost difference, it will follow that $\overline{O}$ is more likely than O given A. At the same time, since this cost difference is larger than the cost difference for C, which is $2(\sqrt{2} - 1)$, the posterior probability of C will be higher than the posterior of A, as shown in the figure.

Planning Background

A Strips planning problem is a tuple $P = \langle F, I, A, G\rangle$ where F is a set of fluents, $I \subseteq F$ and $G \subseteq F$ are the initial and goal situations, and A is a set of actions a with precondition, add, and delete lists $Pre(a)$, $Add(a)$, and $Del(a)$, all subsets of F. A problem P defines a state model whose states s, represented by subsets of F, encode truth valuations; namely, the fluents that are true in s. In this model, the initial state is $s = I$, the goal states are those that include the goals G, and the actions a applicable in a state s are those for which $Pre(a) \subseteq s$, that transform s into $s' = (s \cup Add(a)) \setminus Del(a)$.

A solution or plan for P is an applicable action sequence that maps the initial state into a goal state, and the plan is *optimal* if it has minimum cost. For this, each action a is assumed to have a non-negative cost $c(a)$ so that the cost of an action sequence $\pi = a_1, \ldots, a_n$ is $c(\pi) = \sum c(a_i)$. Unless stated otherwise, action costs are assumed to be 1, so the optimal plans, by default, are the shortest ones.

A number of extensions of the Strips language for representing problems with deterministic actions and full information about the initial state are known. For convenience, we will make use of two such extensions. One is allowing negated fluents like $\neg p$ in goals. The other is allowing *one* conditional effect of the form '$p \rightarrow q$' in some of the actions, where the antecedent p is a single fluent. Both of these constructs can be easily compiled away (Gazen and Knoblock 1997), even if this is not strictly needed. Indeed, state-of-the-art planners compile the negation away but keep conditional effects, as the compilation of *general* conditional effects may be exponential.

Plan Recognition over a Domain Theory

The plan recognition problem given a library for a set $\mathcal{G}$ of goals G can be understood, at an abstract level, as the problem of finding a goal G with a plan π in the library such that π satisfies the observations. In (Ramirez and Geffner 2009), the plan recognition problem over a domain theory is defined instead by replacing the set of plans for G in the library by the set of optimal plans for G. A planning problem P without a goal is called a planning *domain* so that a planning *problem* $P[G]$ is obtained by concatenating a planning domain with a goal G. The *(classical) plan recognition problem* is then defined as follows:

Definition 1. A plan recognition problem *is a triplet* $T = \langle P, \mathcal{G}, O\rangle$ *where* $P = \langle F, I, A\rangle$ *is a planning domain,* $\mathcal{G}$ *is a set of possible goals* G*,* $G \subseteq F$*, and* $O = o_1, \ldots, o_m$ *is the action sequence that has been observed,* $o_i \in A, i \in [1, m]$.

The *solution* to a plan recognition problem is expressed as the subset of goals $G \in \mathcal{G}$ such that some *optimal plan* for $P[G]$ satisfies the observation sequence O. An action sequence $a_1, \ldots, a_n$ satisfies the observation sequence $o_1, \ldots, o_m$ iff it embeds it; i.e. if there is a monotonic function f mapping the observation indices $j = 1, \ldots, m$ into action indices $i = 1, \ldots, n$, such that $a_{f(j)} = o_j$. The resulting subset of goals, denoted as $\mathcal{G}_T^*$, is the *optimal goal set*. This set can be computed exactly by slight modification of existing optimal planners, and can be approximated using satisficing planners. In both cases, the planners are invoked not on the problem $P[G]$, that does not take observations into account, but over a transformed problem whose solutions are the plans for $P[G]$ that satisfy the observations. Before proceeding with the new formulation, we introduce an alternative transformation that is simpler and yields both the plans for $P[G]$ that comply with O and the plans for $P[G]$ that do not. Both will be needed for obtaining the posterior probabilities $P(G|O)$.

Handling the Observations

The new transformation is defined as a mapping of a domain P into a domain P' given the observations O. With no loss of generality, we assume that no action appears twice in O. Otherwise, if a_i and a_k in O refer to the same action a for $i < k$, we make a_k denote another action b that is identical to a except in its name.

Definition 2. *For a domain $P = \langle F, I, A \rangle$ and the observation sequence O, the new domain is $P' = \langle F', I', A' \rangle$ with*

- $F' = F \cup \{ p_a \,|\, a \in O \}$,
- $I' = I$, *and*
- $A' = A$

where p_a is a new fluent, and the actions $a \in A'$ that are in O have an extra effect: p_a when a is the first observation in O, and $p_b \rightarrow p_a$ when b is the action that immediately precedes a in O.

In the transformed domain P', a fluent p_a is made true by an action sequence π if and only if π satisfies the sequence O up to a. In contrast to the transformation introduced in (Ramirez and Geffner 2009), the new transformation does not add new actions, and yet has similar properties.

Let us refer by $G + O$ and $G + \overline{O}$ to the goals that correspond to $G \cup \{p_a\}$ and $G \cup \{\neg p_a\}$, where a is the *last* action in the sequence O. The rationale for this notation, follows from the following correspondence:

Proposition 3. *1) π is a plan for $P[G]$ that satisfies O iff π is a plan for $P'[G + O]$. 2) π is a plan for $P[G]$ that does* not *satisfy O iff π is a plan for $P'[G + \overline{O}]$.*

In words, the goals $G+O$ and $G+\overline{O}$ in P' capture exactly the plans for G that satisfy and do not satisfy the observation sequence O respectively.

If we let $cost(G)$, $cost(G, O)$, and $cost(G, \overline{O})$ stand for the optimal costs of the planning problems $P'[G]$, $P'[G + O]$, and $P'[G + \overline{O}]$ respectively, then the approach in (Ramirez and Geffner 2009) can be understood as selecting the goals $G \in \mathcal{G}$ for which $c(G) = c(G, O)$ holds. These are the goals G for which there is no cost increase when O must be satisfied. The account below focuses instead on the cost difference $c(G, O) - c(G, \overline{O})$, which will allow us to induce a probability distribution over the goals and not just a partition.

As an illustration of the distinction between the cost differences $c(G, O) - c(G)$ and $c(G, O) - c(G, \overline{O})$, consider an example similar to the Noisy Walk, where the agent can only do horizontal and vertical moves at cost 1, and the first move of the agent is to go up (O). If L, M, R are the possible goals and they are all above the initial agent location, complying with the observation will not increase the cost to the goal, and hence $c(G) = c(G, O)$. On the other hand, if only M is located 'right above' the agent position, *not complying with* O, will penalize the achievement of M but not of L or R that have optimal plans where the first action is not to go up. The result in the account below is that, if the priors of the goals M, L, and R are equal, the posterior of M will be higher than the posteriors of L and R. This is because the goal M predicts the observation better than either L or R.

Probabilistic Plan Recognition

In order to infer posteriors $P(G|O)$, we need first information about the *goal priors* in the problem description:

Definition 4. *A* probabilistic plan recognition problem *is a tuple $T = \langle P, \mathcal{G}, O, Prob \rangle$ where P is a planning domain, $\mathcal{G}$ is a set of possible goals G, O is an* observation sequence, *and $Prob$ is a probability distribution over $\mathcal{G}$.*

The posterior probabilities $P(G|O)$ will be computable from Bayes Rule as:

$$P(G|O) = \alpha \, P(O|G) \, P(G) \tag{1}$$

where α is a normalizing constant, and $P(G)$ is $Prob(G)$. The challenge in this formulation is the definition of the likelihoods $P(O|G)$ that express the probability of observing O when the goal is G. A rationality postulate softer than the one adopted in (Ramirez and Geffner 2009) is that G is a better predictor of O when the cost difference $c(G, O) - c(G, \overline{O})$ is smaller. Indeed, G is a *perfect predictor* of O when all the plans for G comply with O, as in such a case the cost difference is $-\infty$.

Assuming a Boltzmann distribution and writing $exp\{x\}$ for e^x, we define then the likelihoods as:

$$P(O|G) \stackrel{\text{def}}{=} \alpha' \, exp\{-\beta \, c(G, O)\} \tag{2}$$

$$P(\overline{O}|G) \stackrel{\text{def}}{=} \alpha' \, exp\{-\beta \, c(G, \overline{O})\} \tag{3}$$

where α' is a normalizing constant, and β is a positive constant. If we take the ratio of these two equations, we get

$$P(O|G)/P(\overline{O}|G) = exp\{-\beta \, \Delta(G, O)\} \tag{4}$$

where $\Delta(G, O)$ is the cost difference

$$\Delta(G, O) = c(G, O) - c(G, \overline{O}) \; . \tag{5}$$

The equations above, along with the priors, yield the posterior distribution $P(G|O)$ over the goals. In this distribution, when the priors are equal, it is simple to verify that the most likely goals G will be precisely the ones that minimize the expression $cost(G, O) - cost(G, \overline{O})$.

Assumptions Underlying the Model

Equations 2–3 are equivalent to Equation 5; the former imply the latter and can be derived from the latter as well. The equations, along with Bayes Rule, define the probabilistic model. The assumptions underlying these equations, are thus the assumptions underlying the model. The first assumption in these equations is a very reasonable one; namely, that when the agent is pursuing a goal G, he is more likely to follow cheaper plans than more expensive ones. The second assumption, on the other hand, is less obvious and explicit, and it's actually an *approximation*; namely, that the probability that the agent is pursuing a plan for G is *dominated* by the probability that the agent is pursuing one of the most likely plans for G.

Indeed, it is natural to define the likelihoods $P(O|G)$ as as the sum

$$P(O|G) = \sum_{\pi} P(O|\pi) \cdot P(\pi|G) \tag{6}$$

with π ranging over all possible action sequences, as the observations are independent of the goal G given π. In this expression, $P(O|\pi)$ is 1 if the action sequence π embeds the sequence O, and 0 otherwise. Moreover, assuming that $P(\pi|G)$ is 0 for action sequences π that are not plans for G, Equation 6 can be rewritten as:

$$P(O|G) = \sum_{\pi} P(\pi|G) \tag{7}$$

where π ranges now over the plans for G that comply with O. Then if a) the probability of a plan π for G is proportional to $exp\{-\beta\, c(\pi)\}$ where $c(\pi)$ is the cost of π, and b) the sum in (7) is dominated by its largest term, then the model captured by Equations 2–3 follows once the same reasoning is applied to $\overline{O}$ and the normalization constant α' is introduced to make $P(O|G)$ and $P(\overline{O}|G)$ add up to 1. The approximation b) is an order-of-magnitude approximation, of the type that underlies the 'qualitative probability calculus' (Goldszmidt and Pearl 1996). The result of this implicit approximation is that *the probabilities corresponding to different plans for the same goal are not added up.* Thus, in the resulting model, if there are four different plans for G with the same cost such that only one of them is compatible with O, $\overline{O}$ will be deemed to be as likely as O given G, even if from a) alone, $\overline{O}$ should be three times more likely than O. The approximation b) is reasonable, however, when cheaper plans are much more likely than more expensive plans, and the best plans for $G + O$ and $G + \overline{O}$ are unique or have different costs. In any case, these are the assumptions underlying the probabilistic model above, which *yields a consistent posterior distribution over all the goals, whether these conditions are met or not.*

Experimental Results

We have tested the formulation above empirically over several problems, assuming equal goal priors and computing costs using both optimal and satisficing planners. The first class of problems is from (Ramirez and Geffner 2009) and involves variants of well known planning benchmarks: BLOCK WORDS, IPC–GRID and LOGISTICS. The other class of problems includes INTRUSION DETECTION, CAMPUS and KITCHEN. These problems have been derived manually from the plan libraries given in (Geib and Goldman 2002; Bui et al. 2008; Wu et al. 2007).

The planners are HSP^*_F, an optimal planner (Haslum 2008), and LAMA (Richter, Helmert, and Westphal 2008), a satisficing planner that is used in two modes: as a *greedy* planner that stops after the first plan, and as an *anytime* planner that reports the best plan found in a given time window. The times for HSP^*_F, anytime LAMA, and greedy LAMA were limited to four hours, 240 seconds, and 120 seconds respectively, *per plan recognition problem*. Each PR problem involves $|\mathcal{G}|$ possible goals, requires the computation of $2|\mathcal{G}|$ costs, and hence, involves running the planners over $2|\mathcal{G}|$ *planning problems*. Thus, on average, each of the planners needs to solve this number of planning problems in the given time window. The experiments were conducted on a dual-processor Xeon 'Woodcrest' running at 2.33 GHz and 8 Gb of RAM. All action costs have been set to 1.

The results over the six domains are summarized in Table 1, whose rows show averages over 15 PR problems obtained by changing the goal set $\mathcal{G}$ or the observation sequence O. The observation sequences are generated by sampling hidden optimal plans for a hidden goal in the first four domains, and by sampling hidden suboptimal plans for a hidden goal, in the others. The number of observations in each row correspond to the percentage of actions sampled from the hidden plan: 10%, 30%, 50%, 70%, and 100% as shown. For each domain, the average size of $\mathcal{G}$ is shown. The columns for HSP^*_F express the 'normative' results as derived from the optimal costs. The column T in all cases shows the *average time* per plan recognition problem. These times are larger for the optimal planner, approach the 240 seconds time window for the anytime planner, and are lowest for the greedy planner. For example, the time of 43.19 seconds in the third row for greedy LAMA, means that $43.19s$ was the avg time over the 15 Block Word *plan recognition problems* that resulted from sampling 50% of the observations. Since the problem involves 20 possible goals, and for each goal, the planning problems with goals O and $\overline{O}$ are solved, in the 43.19 seconds, 2×20 classical planning problems are solved.

The column L displays average optimal plan length, while the columns Q and S provide information about the quality of the solutions by focusing on the goals found to be the most likely. Q equal to 0.9 means that the hidden goal used to generate the observations was among the goals found to be the most likely 90% of the time. S equal to 2.4 means that, on average, 2.4 of the goals in $\mathcal{G}$ were found to be the most likely. It is important to notice that it is not always the case that the hidden goal used to generate the observations will turn out to be the most likely given the observations, even when O is compatible with an optimal plan for G. Indeed, if there are optimal plans for G that do not agree with O, and there is another achievable goal G' that has not such optimal plans, then in the formulation above, $P(G'|O)$ will be higher than $P(G|O)$. This is entirely reasonable, as G' is then a perfect predictor of O, while G is not.

		HSP$^*_f$				LAMA (240s)			Greedy LAMA		
Domain	O	T	Q	S	L	T	Q	S	T	Q	S
BLOCK WORDS $\|\mathcal{G}\| = 20$	10	1184.23	1	6	10	228.04	0.75	4.75	52.79	0	1.67
	30	1269.31	1	3.25	11	239.59	1	3	53.01	0.5	2
	50	1423.05	1	2.23	11	241.77	1	2.23	53	0.54	1.23
	70	1787.67	1	1.27	12	241.53	1	1.27	53.06	0.73	1.2
	100	2100.21	1	1.13	12	241.51	1	1.13	53.47	0.73	1.07
EASY IPC GRID $\|\mathcal{G}\| = 7.5$	10	73.38	0.75	1.38	15	22.15	0.75	1.38	3.96	0.75	1.38
	30	155.47	1	1	17	64.63	1	1	5.38	1	1.08
	50	202.69	1	1	17	71.77	1	1	9.2	1	1
	70	329.64	1	1	20	92.84	1	1	11.23	1	1
	100	435.6	1	1	18	90.22	1	1	13.07	1	1
INTRUSION DETECTION $\|\mathcal{G}\| = 15$	10	26.29	1	1.8	18	62.38	1	1.8	3.69	1	2.2
	30	73.08	1	1.13	19	142.63	1	1.13	4.09	1	1.13
	50	103.58	1	1	20	194.55	1	1	4.44	1	1
	70	188.44	1	1	21	223.97	1	1	4.96	1	1
	100	179.41	1	1	21	224.96	1	1	5.94	1	1
LOGISTICS $\|\mathcal{G}\| = 10$	10	120.94	0.9	2.3	21	215.32	0.9	2.3	4.35	0.6	1.8
	30	1071.91	1	1.07	22	236.29	1	1.07	4.55	0.87	1.13
	50	813.36	1	1.2	23	238.87	1	1.2	5.37	1	1.2
	70	606.87	1	1	24	243.38	1	1	6.29	1	1
	100	525.44	1	1	24	247.04	1	1	8.34	1	1
CAMPUS $\|\mathcal{G}\| = 2$	10	0.67	0.93	1.33	10	0.97	0.93	1.33	0.74	0.67	1.27
	30	0.92	1	1	11	1.13	1	1	0.74	0.8	1.07
	50	1.11	1	1	11	1.31	1	1	0.77	0.8	1.13
	70	1.41	1	1	11	1.63	1	1	0.8	0.8	1
	100	1.56	1	1	11	1.84	1	1	0.82	1	1.2
KITCHEN $\|\mathcal{G}\| = 3$	10	77.85	0.88	1.25	11	80.74	0.88	1.25	1.55	0.88	1.25
	30	144.58	0.93	1.21	11	80.82	0.93	1.21	0.67	0.93	1.21
	50	218.51	1	1.33	11	80.86	1	1.33	0.71	1	1.27
	70	245.88	1	1.2	11	80.86	1	1.2	0.73	1	1.47
	100	488	1	1.47	12	81.16	1	1.4	0.82	1	1.6

Table 1: Evaluation with an optimal and two satisficing planners. Each row describes averages over 15 plan recognition problems. The columns stand for % of actions in hidden plan sampled, avg time in seconds for each complete plan recognition problem (T), avg quality measuring fraction of problems where hidden goal is among the most likely (Q), avg number of most likely goals (S).

The table shows that the approach has good precision and scales up well, with anytime LAMA over the 250 seconds time window, matching the quality of the exact account in much less time, and greedy LAMA not lagging far behind in quality while often taking more than an order-of-magnitude less time. A more detailed description of the problems follows.[1]

BLOCKS WORD is Blocks World with six blocks labeled with letters and arranged randomly, that must be ordered into a single tower to spell one of 20 possible words (goals). As in the other domains, each possible goal is *conjunctive*, and involves in this case, 6 fluents. The task is to recognize the word from the observations. IPC-GRID is about an agent that moves in a grid and whose task is to transport keys from some cells to others. The locations may be locked, however, and for entering the cell to pick up a key, another key may be needed. In this problem, the agent can hold one key at a time, but the goals are conjunctive and involve positioning several keys. The number of possible goals in $\mathcal{G}$ ranges from 5 to 10. LOGISTICS is about carrying packages among locations using planes and trucks. In these problems, the number of packages is 6, the number of trucks ranges from 1 to 2 per city, and there is just one plane. $|\mathcal{G}|$ is 10, and each possible goal $G \in \mathcal{G}$ specifies the destination for some or all the packages.

In INTRUSION DETECTION (Geib and Goldman 2002), the agent is a hacker who might try to either gain access, vandalize, steal information, or perform a combination of these *attacks* on a set of *servers*. Actions available to the hacker range from port reconnoitering to downloading files from the server. Some actions require one or more actions being done before. We modeled the plan library graphs in Strips with edges $A \rightarrow B$ in the graph being mapped into Strips actions A that add a precondition of the action B (a more general translation of libraries into Strips, can be found in (Lekavý and Návrat 2007)). Each type of attack on a particular server becomes a single fluent, which we use to define the set of possible goals $\mathcal{G}$. We include in this set all conjunctive goals standing for the combinations of up to *three attacks* on *three different servers*. Since there are 9 different actions to perform in each machine, and we consider up to 20 machines, the total number of Strips actions is 180.

The CAMPUS domain from (Bui et al. 2008) is about finding out the activity being performed by a student by tracking his movements. The plan library graphs were converted into Strips theories as indicated above, with the provision that each activity (high or low level) is performed at certain locations only. The only actions observed are changes in location, from which the top level goal must be inferred. In this case, $|\mathcal{G}|$ is two, and there are 11 different activities and locations, resulting in 132 Strips actions.

Finally, in KITCHEN, the possible activities, translated into goals, are three: preparing dinner, breakfast, and lunch (Wu et al. 2007). Actions are used to encode low level activities such as "make a toast" or "take bread", and higher-level activities such as "make cereal". The domain features several plans or methods for each top activity. The observation sequences contain only "take" and "use" actions; the first involves taking an object, the second, using an appliance. The other actions or activities are not observable. Our PR tasks involve 32 "take" actions (for 32 different objects), 4 "use" actions (for 4 different appliances), and 27 Strips actions encoding the higher level activities that are not observable. The total number of actions is thus 63.

Discussion

We have extended the generative approach to plan recognition introduced recently in (Ramirez and Geffner 2009), for inferring meaningful probability distributions over the set of possible goals given the observations. The probabilities $P(G|O)$ are determined by the goal priors and the likelihoods $P(O|G)$ obtained from the difference in costs arising from achieving G under two conditions: complying with the observations, and not complying with them. These costs, as in the previous formulation, are computed using optimal and satisficing planners. We have shown experimentally that this

[1]The problems and code used in the evaluation can be found at `https://sites.google.com/site/prasplanning`.

approach to 'classical' plan recognition renders good quality solutions, is flexible, and scales up well. Two additional benefits over the previous formulation are that the classical planners can be used off-the-shelf, and that the approach does not assume that agents are perfectly rational and hence only follow plans with minimal cost. Independently of this work, Baker *et. al.* have also pursued recently a generative approach to plan recognition in the setting of Markov Decision Processes (Baker, Saxe, and Tenenbaum 2009). This approach, however, assumes that the optimal Q-value function $Q_G(a, s)$ can be computed and stored for all actions, states, and goals. In addition, the observations refer to complete plan prefixes with no gaps.

Most other recent approaches to probabilistic plan recognition are not generative and make use of suitable libraries of plans or policies (Bui 2003; Pentney et al. 2006; Geib and Goldman 2009). A basic difficulty with these approaches concerns the recognition of joint plans for achieving goal conjunctions. Joint plans come naturally in the generative approach from conjunctive goals, but are harder to handle in library-based methods. Indeed, semantically, it is not even clear when a plan for two goals can be obtained from the combination of plans for each goal in isolation. For this, in general, a domain theory is required.

From a computational point of view, a current popular approach is the mapping of the activity recognition problem into an evidential reasoning task over a Dynamic Bayesian Network, where approximate algorithms such as Rao-Blackwellised particle filtering are used (Doucet et al. 2000). The advantage of this approach is its expressive power: while it can only recognize a fixed set of prespecified policies, it can handle stochastic actions and sensors, and incomplete information. On the other hand, the quality and scalability of these approaches for long planning horizons is less clear. Indeed, the use of time indices to tag actions, observations, and hidden states is likely to result in poor performance when the time horizon required is high. This limitation arises also in the SAT approach to classical planning (Kautz and Selman 1999), and would arise from the use of Max Weighted-SAT methods for plan recognition as well, as similar temporal representations would be required.

The proposed formulation is less vulnerable to the size of domain theories and does not involve temporal representations or horizons. On the other hand, it is limited to the 'classical' setting where actions are deterministic and the initial state is fully known. In the future, we want to analyze further the assumptions underlying the proposed model, and to tackle the plan recognition problem in richer settings, such as contingent and POMDP planning, also using off-the-shelf solvers and suitable translations,

Acknowledgments. H. Geffner is partially supported by grant TIN2009-10232, MICINN, Spain.

References

Baker, C. L.; Saxe, R.; and Tenenbaum, J. B. 2009. Action understanding as inverse planning. *Cognition* 113(3):329–349.

Bui, H. H.; Phung, D.; Venkatesh, S.; and Phan, H. 2008. The hidden permutation model and location-based activity recognition. In *Proc. AAAI-08*.

Bui, H. 2003. A general model for online probabilistic plan recognition. In *Proc. IJCAI-03*, 1309–1318.

Cohen, P. R.; Perrault, C. R.; and Allen, J. F. 1981. Beyond question answering. In Lehnert, W., and Ringle, M., eds., *Strategies for Natural Language Processing*. LEA.

Doucet, A.; de Freitas, N.; Murphy, K.; and Russell, S. 2000. Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proc. UAI-2000*, 176–183.

Gazen, B., and Knoblock, C. 1997. Combining the expressiveness of UCPOP with the efficiency of Graphplan. In *Proc. ECP-97*, 221–233. Springer.

Geib, C. W., and Goldman, R. 2002. Requirements for plan recognition in network security systems. In *Proc. Int. Symp. on Recent Advances in Intrusion Detection*.

Geib, C. W., and Goldman, R. P. 2009. A probabilistic plan recognition algorithm based on plan tree grammars. *Artificial Intelligence* 173(11):1101–1132.

Goldszmidt, M., and Pearl, J. 1996. Qualitative probabilities for default reasoning, belief revision, and causal modeling. *Artificial Intelligence* 84(1-2):57–112.

Haslum, P. 2008. Additive and Reversed Relaxed Reachability Heuristics Revisited. Proceedings of the 2008 Internation Planning Competition.

Huber, M. J.; Durfee, E. H.; and Wellman, M. P. 1994. The automated mapping of plans for plan recognition. In *Proc. UAI-94*, 344–351.

Kautz, H., and Allen, J. F. 1986. Generalized plan recognition. In *Proc. AAAI-86*, 32–37.

Kautz, H., and Selman, B. 1999. Unifying SAT-based and Graph-based planning. In Dean, T., ed., *Proceedings IJCAI-99*, 318–327. Morgan Kaufmann.

Lekavý, M., and Návrat, P. 2007. Expressivity of Strips-like and HTN-like planning. In *Proc. 1st KES Int. Symp.KES-AMSTA 2007*, 121–130.

Lesh, N., and Etzioni, O. 1995. A sound and fast goal recognizer. In *Proc. IJCAI-95*, 1704–1710.

Pentney, W.; Popescu, A.; Wang, S.; Kautz, H.; and Philipose, M. 2006. Sensor-based understanding of daily life via large-scale use of common sense. In *Proc. AAAI-06*.

Ramirez, M., and Geffner, H. 2009. Plan recognition as planning. In *Proc. 21st Intl. Joint Conf. on Artificial Intelligence*, 1778–1783. AAAI Press.

Richter, S.; Helmert, M.; and Westphal, M. 2008. Landmarks revisited. In *Proc. AAAI*, 975–982.

Wu, J.; Osuntogun, A.; Choudhury, T.; Philipose, M.; and Rehg, J. 2007. A scalable approach to activity recognition based on object use. In *Proc. of ICCV-07*.

Yang, Q. 2009. Activity Recognition: Linking low-level sensors to high-level intelligence. In *Proc. IJCAI-09*, 20–26.