Domain Independent Goal Recognition

David Pattison
Department of Computer and Information Science
University of Strathclyde, Glasgow G1 1XH, UK
david.pattison@cis.strath.ac.uk

Abstract

Goal recognition is generally considered to follow *plan* recognition. The plan recognition problem is typically defined to be that of identifying which plan in a given library of plans is being executed, given a sequence of observed actions. Once a plan has been identified, the goal of the plan can be assumed to follow. In this work, we address the problem of goal recognition directly, without assuming a plan library.

We present a formalisation of the problem and motivate its interest, before describing some simplifying assumptions we have made to arrive at a first implementation of a goal recognition system, AUTOGRAPH. We discuss the techniques employed in AUTOGRAPH to arrive at a tractable approximation of the goal recognition problem and show results for the system we have implemented, before discussing future research possibilities and their impact.

1. Introduction

Goal Recognition (GR) is the process of inferring an agent's end goals given a series of observed actions. This is clearly related to the *Plan Recognition* (PR) problem which aims to also find the plan being executed. *Planning* is simply the generation of these plans in an efficient and sensible manner. Yet despite both being based on *actions*, *states* and *goals*, and effectively mirroring one another, advances in research have rarely overlapped.

Previous work has often focused on a single application of the recognition problem, such as identification of human goals through observation of behaviour (Huntemann et al. 2008), giving speech/text context (Gorniak and Roy 2005) or responding with natural dialogue (Mott, Lee, and Lester 2006). These have all resulted in systems and algorithms that lack generality or widespread application.

AUTOGRAPH (AUTOmatic Goal Recognition with A Planning Heuristic), is a new approach to Goal Recognition which makes use of Planning techniques. The system uses a standard planning domain model, avoiding the construction of a goal/plan library.

2. Motivation and Prior Approaches

Plan and Goal Recognition problems are motivated by the desire to anticipate the actions or objectives of an agent that is being observed. There are many situations in which this could be useful, including detection and prevention of crime, in teaching, in monitoring the elderly or infirm in their own homes, in military operations and in games. In computer games, intelligent responses to human player activity depend on recognising what that activity might be. Creating a believable and responsive environment that allows players to participate in a truly immersive experience requires that computer controlled agents react to human players with plausible levels of understanding of the human players' actions. This context, in particular, motivates two assumptions underlying our work: first, that the actions are fully observable (game software mediates every action on behalf of the players) and, second, that we are interested in identifying goals as early as possible during the execution of the plan.

Kautz (1987) defines the plan recognition problem as minimising the number of top-level, hierarchical plans which explain a sequence of observed actions. Plans were taken from a *plan graph* and every action is assumed to be *relevant* to the plan being executed. The library containing known, valid plans has remained a key element of plan recognition ever since. This structure presents several drawbacks such as the time, effort and space required to construct it and its inevitable incompleteness and irrelevant content. AUTOGRAPH attempts to address these problems in three areas: *Completeness*, *Scalability* and *Domain Independence*.

Completeness: It is impossible to generate and store every valid plan in a library for non-trivial problems. Previous work has often made use of tree-like structures to represent a large number of plans efficiently but cannot hold *all* possible plans or goals. In our work, any conjunction of literals may form a hypothesis.

Scalability: The scaling behaviour of plan recognition systems is highly dependent on library sizes. This and the previous problems combine to create a tension between scalability and completeness.

Domain Independence: Generating plan libraries is time-consuming and restricts application to domains for which libraries are available.

We note that these objectives have been tackled previously, but never in conjunction. Blaylock and Allen (2003) and Lesh and Etzioi (1995; 1997) have also explored adapting recognition to a previously unseen plan but require that their systems be trained first and/or have access to an explicit representation of the goal set.

Hong (2001) and more recently Ramírez and Geffner (2009) have also applied Planning techniques to recognition, but both still require an enumerated set of goals.

3. Problem Definition

We start with the same framework that is used in classical planning, based on a propositional action model structure. A goal recognition problem is based on a standard planning problem (the facts, actions and initial state). Of course, the goal recognition problem does not contain a goal specification — the problem is to find this specification.

Definition 1. *Goal Recognition Problem Base*
A goal recognition problem base is a triple $\langle F, A, I\rangle$, where F is a set of primitive (propositional) facts, A is a set of actions and $I \subseteq F$ is the initial state for the problem. Each action $a \in A$ is a triple $\langle pre_a, add_a, del_a\rangle$, where $pre_a, add_a, del_a \subseteq F$ are the preconditions, add effects and delete effects of a, respectively.

In addition to the base, a goal recognition problem requires observations: a sequence of actions. We assume that all actions and states are fully observable, but we want to identify the goals as early as possible during execution of the plan. Before we define the goal recognition problem, however, we briefly consider the nature of the solutions we seek and the implications this has on the problem itself. Our expectation is that we should be presented with a goal recognition problem base and a series of actions, with the objective being to identify the target goals of the agent performing the actions. We assume that the agent actually has a target and is not simply executing actions at random.

In general, there are many goal sets consistent with a sequence of observed actions, ranging from the possibility that the most recent state was in fact the goal state to the possibility that there are many goals towards which the agent has not yet even begun to act. However, these possibilities are not all equally likely: in most domains there is a clear bias towards certain kinds of goals. This motivates the following definition:

Definition 2. *Goal Hypothesis and Goal Hypothesis Space*
Given a goal recognition problem base, G, with facts F, a goal hypothesis for G is a probability distribution over subsets of F reachable from the initial state using actions in G. The goal hypothesis space for G, $\mathbb{H}$, is the set of all goal hypotheses for G.

Definition 3. *Goal Recognition Problem*
A goal recognition problem is a triple, $\langle G, H_I, (o_1, o_2, ..., o_n)\rangle$, where G is a goal recognition problem base, H_I is an initial goal hypothesis and $(o_1, ..., o_n)$ is the sequence of actions observed one-by-one during the problem.

Each observation in a goal recognition problem updates the hypothesis space, so that candidate goals that are further away from the new state than the previous state are assigned an updated probability of 0, while the remaining probability mass is re-normalised across the other states.

Unfortunately, explicit representation of $\mathbb{H}$ for anything other than trivial problems is impossible due to its exponential size. We therefore introduce an approximation of the space which is tractable, but at the price that we cannot accurately represent all possible goal hypotheses.

Definition 4. *Approximate Goal Hypothesis*
An nth order approximation to a goal hypothesis, H, is a goal hypothesis, $\hat{H}$, where $\hat{H}(f) = H(f)$ when $|f| \leqslant n$ and, $\hat{H}(f) = \min_{x\in f} \hat{H}(f \setminus \{x\}) \cdot \hat{H}(\{x\})/N$, where N is an appropriate normalising factor to ensure that $\hat{H}$ is a probability distribution.

An approximate goal hypothesis is not necessarily a member of the same goal hypothesis space as the goal hypothesis it approximates, because the approximation can assign non-zero probabilities to unreachable sets of facts. Identifying unreachable sets is as hard as planning, so allowing these sets to be assigned non-zero values is a useful efficiency measure. The method by which probabilities are combined in the recursive extension of the approximation to the whole space of possible goals is somewhat arbitrary and alternative approximations are certainly possible. In our current work we only consider 1st order approximations, so the probability of sets of facts is the product of the probabilities of the individual facts they include. This is equivalent to assuming that the individual goals have independent probabilities of appearing. Although 1st order approximations are poor in domains where goals are strongly correlated, in many domains we see goals falling into independent selections of states of a collection of objects (such as packages in a delivery domain).

This independence assumption clearly does not hold for all domains, for example BLOCKSWORLD problems often have the same numbers of goal and initial-state literals. We currently focus on problems which do exhibit this property, although we also consider the performance of the approximation on other benchmark domains.

Within the framework we have now defined, it is apparent that each successive observation implies an update of the current goal hypothesis reducing the probabilities of reachable facts that are subsets of those states which are now no closer in the state than the prior state. However, it is impractical to identify exactly which states these are. Furthermore, the assumption that the agent that is being observed has the capability to identify the shortest path to its goal, without error, is unreliable. For this reason, we work with 1st order approximations and update by reducing the probability associated with *facts* that get further away following observed actions and increasing the probabilities of *facts* that get closer.

4. Recognition without Libraries

AUTOGRAPH performs goal recognition in four stages: *Analysis*, wherein the problem is instantiated and analysed to reveal useful aspects of the domain; *Observation*, in which a single, ordered action is fed into the recogniser and the current state updated to reflect its effects; *Intermediate Hypothesis Generation*, in which a single hypothesis is produced after each observation[1]; and, lastly, a *Final Hypothesis* is generated once the plan is known to have finished.

4.1 Analysis

Domain analysis can provide rich information to aid subsequent search (Fox and Long 1998; Helmert 2004), lowering search time and shortening plan length. We apply relevant prior research to GR and develop new techniques that allow the recogniser to make more informed hypotheses.

Problem Representation We use domains encoded in PDDL2.1 (Fox and Long 2003) and then apply Helmert's translator (2009) to translate these into a SAS+ formalism. Two key products of this translation process are Domain Transition Graphs (DTGs) and a Causal Graph (CG), both of which encode aspects of the original PDDL problem in another form. We use both the PDDL and SAS+ representations of the problem during analysis, as they can each reveal aspects of the domain that aid in recognition. The Causal Graph reveals how objects influence others within the domain through actions: of particular interest are the *leaf nodes*, corresponding to objects with no influence on others. Should a causal graph contain leaf nodes, any fact containing a leaf variable that is **not** true in the initial state can be seen as a likely goal, as it can play no role other than to be altered.

Predicate Partitioning Geib (Geib 2009) proposed the concept of *plan heads* in Plan Recognition as a way to highlight important plan actions and allowing lazy commitment to plans, resulting in faster runtimes. We adapt this idea for GR through the concept of *predicate partitioning*. By automatically classifying propositions into mutually-exclusive sets it can often become clear which are more or less likely to be goals. For example, in a standard Logistics problem it is unlikely that the goal will be to simply have a

[1] If the plan has further actions to be observed, steps 2 and 3 are repeated until this is no longer the case.

package inside a truck and far more likely that it must be delivered to a warehouse. Facts can be placed in the following sets through analysis of the two domain representations, which can then be applied to the initial probability distribution. Formal definitions of each partition have been excluded due to space limitations.

Definition 5. *Predicate Partitions– A fact f is:*

1. ***strictly activating*** *Cannot be removed if present in the initial state and therefore extremely unlikely to be goals.*
2. ***unstably activating*** *Can be deleted by at least one action, but once removed from the current state cannot be re-added. Once deleted they can be removed from future hypotheses.*
3. ***strictly terminal*** *Do not appear as a precondition, and once added these cannot be removed, meaning they must appear in the final state. Thus they are highly likely to be goals.*
4. ***unstably terminal*** *Unlike strictly terminal facts, these can be removed once they have been added, but they are never used as preconditions to any actions.*
5. ***waypoint*** *It is common for problems to involve transforming objects through a chain of related states, all defined by the same predicate. Any facts located within this predicate-chain (excluding end-points) are assigned a low initial probability.*
6. ***transient*** *These facts are transitions between facts of the same predicate. It is unlikely that the goal will be to leave the object within this transient state.*
7. ***binary*** *Facts which can only transition between 2 values are assigned low initial probabilities as it is difficult to assess which of them might be relevant to the goal.*

In addition to these sets, a further *neutral* set is defined, containing all facts that have not been partitioned into one of the above sets.

The population of the various partitions is dependent on the domain being analysed. For example the ROVERS domain populates 5 partitions, while others such as ZENOTRAVEL largely categorise facts in the *waypoint* and *transient* sets. The populations of these partitions are used during construction of the *initial-probability distribution*.

Unhelpful Facts While Helmert's SAS+ translation also approximates the set of all *reachable* facts, it is likely that some will never appear as a goal. We begin to reduce the set of facts by first observing that it is extremely unlikely that a problem will be considered a planning problem if its goals can be achieved in a single step, since this could be achieved by purely greedy action selection. Therefore, any action applicable in the initial state is considered *unhelpful* and its effects are assigned negligible probability in the initial hypothesis. Additionally, if the domain contains *strictly-terminal* facts, we assign negligible probability to the preconditions of any action which achieves them by reasoning that the enabling conditions for achievement of a *strictly-terminal* fact are very unlikely to be goals instead of the terminal itself.

Initial Probability Distribution Once the analysis phase has been completed, each fact f in the approximate hypothesis space $\mathcal{H}$ can be assigned an initial probability. This figure is dependent on which, if any, of the previous domain analysis criteria the fact has met.

4.2 Execution

Once the domain has been analysed and the initial goal-space populated, plan observation can begin. After each observation we record the *heuristic estimate* to each fact in the *approximate hypothesis space*.

By observing the estimated distance to each fact after action observations, it is possible to determine those which are being moved towards and away from. Each fact which has a lower heuristic estimate at time t than it did at $t-1$ has its probability of being a goal increased, while those which now have a larger estimate have their probability set to 0. Facts whose estimate remains unchanged do not have their probability updated as they may be goals which have been achieved at time t.

4.3 Hypothesis Generation

By using a 1st order approximation of a goal hypothesis we rely on goal sets being small. However, it would be naïve to assume that all domains only contain a single literal as their goal. We therefore construct an *intermediate greedy hypothesis* h_i from the approximate goal hypothesis constructed at each timestep i, representing the single most likely goal of the agent being observed.

To produce this set, facts are considered in mutually exclusive clusters (the sets that make up the nodes in a single DTG). The fact with highest probability within each cluster is selected, provided it has probability higher than a specified threshold (this eliminates highly unlikely candidates from the set).

Definition 6. *Greedy-Hypothesis*
Given an aptproximate hypothesis space $\mathcal{H}$, a ***greedy-hypothesis*** *h_{gr} is the set of facts with the highest probabilities above a base threshold, T_{min}, with ties broken randomly. If a fact f is chosen, then all facts that are mutex with it will not be added to h_{gr}.*

4.4 Final Hypothesis

Once the plan is known to have terminated and the final state is known, a more accurate final hypothesis can be produced. This is simpler than generating an *intermediate hypothesis* since $G \subseteq S_n$, and the state is certainly mutex-free. Along with the final probability distribution, this can produce a very accurate goal hypothesis without considering fact probabilities.

5. Initial Results

We now present empirical results of several tests performed on the techniques presented previously. While others have previously expressed a desire for plan and goal recognition to have a standard evaluation method (Carberry 2001), there is still no agreement on standard benchmarks. Therefore, we have used classical planning benchmarks as an alternative. The system is evaluated using *precision and recall*, a technique used to score database document-retrieval which has also previously been applied to a GR context (Blaylock and Allen 2006), where the number of required facts in each hypothesis is the *precision* and the number of correct facts is the *recall*.

We have tested the system using the *Max* (h_{max}), *FF* (h_{ff}) and *Causal Graph* (h_{cg}) heuristics (Bonet and Geffner 2001; Hoffmann and Nebel 2001; Richter, Helmert, and Westphal 2008) in order to determine how this choice affects performance.

5.1 Intermediate Hypothesis Results

The results of precision versus recall for *intermediate hypotheses* over all domains and heuristics can be seen in Table 5..1 Also included are the average precision and recall over all problems using h_{ff} at various timepoints. These latter results show *heuristic convergence* as precision and recall increase at over the course of plan execution.

Perhaps of most interest is that there is no clear leader in terms of heuristic chosen to generate estimates. While h_{cg} has the highest overall P+R results, this is primarily caused by the results of ROVERS, which contains more tests than other domains. The normalised results show the difference between this and h_{max} is only 0.06 and 0.03 for precision and recall respectively.

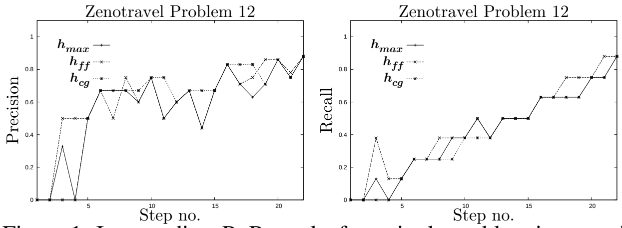

Figure 1: Intermediate P+R results for a single problem instance in ZENOTRAVEL.

Domain	h_{max}	h_{ff}	h_{cg}
Depots	**0.22 / 0.3**	0.22 / 0.28	0.22 / 0.28
Driverlog	0.42 / 0.32	**0.47 / 0.32**	**0.47 / 0.32**
Rovers	0.78 / 0.54	0.82 / 0.5	**0.86 / 0.56**
Zenotravel	0.46 / 0.32	**0.49 / 0.33**	0.48 / 0.31
Storage	**0.19 / 0.42**	**0.22 / 0.39**	**0.22 / 0.39**
Total	0.54 / 0.43	0.58 / 0.4	**0.6 / 0.43**

Domain	h_{ff} 25%	h_{ff} 50%	h_{ff} 75%	h_{ff} 100%
Depots	0.15 / 0.07	0.2 / 0.21	0.34 / 0.5	0.52 / 0.93
Driverlog	0.43 / 0.12	0.56 / 0.29	0.72 / 0.5	0.94 / 0.73
Rovers	0.62 / 0.22	0.81 / 0.42	0.93 / 0.66	0.96 / 1
Zenotravel	0.3 / 0.1	0.42 / 0.26	0.61 / 0.48	0.81 / 0.7
Storage	0.17 / 0.14	0.29 / 0.49	0.27 / 0.64	0.3 / 0.91
Average	0.33 / 0.13	0.46 / 0.33	0.57 / 0.56	0.7 / 0.85

Table 1: The total normalised intermediate precision and recall results for each heuristic, and the average precision and recall for h_{ff} over all problems at 25%, 50%, 75% and 100% plan completion.

5.2 Final Hypothesis Results

Once the plan being observed is known to have finished, the *final hypothesis* can be generated. Knowing that the goal is definitely contained within the final state increases hypothesis accuracy greatly, as we know $G \subseteq S_F$. Table 2 shows the total normalised P+R values for the final hypotheses in each problem. We note that the seeming indifference of heuristic choice is further reinforced by these results, as all three heuristics produce identical final P+R results.

With the exception of DEPOTS and STORAGE, the recogniser produces highly accurate hypotheses for all problems in terms of both precision and recall. ROVERS shows particularly accurate results due to the presence of *strictly-terminal* facts, which results in a perfect score for recall, and 95% average for precision.

In the case of DEPOTS and STORAGE, precision scores average only 52% and 32% respectively. This is caused by the large number of facts which become true during execution of a typical plan, which along with a small goal set combine to form a large hypothesis with extraneous facts. For instance, the location of certain trucks is often not a required goal in DEPOTS, but will be put forward as a goal because trucks will stop moving once its last package has been delivered to its destination.

6. Conclusions and Future Work

We have presented AUTOGRAPH, a new method of tackling Goal Recognition by applying Planning technology. The approach and empirical evidence presented has successfully shown that libraries are not required to achieve online recognition of an agent's activities.

Domain	Depots	Driverlog	Rovers	Zenotravel	Storage
Precision	0.52	0.94	0.96	0.86	0.32
Recall	0.93	0.73	1	0.7	0.95

Table 2: Normalised total values for precision and recall values associated with the final hypothesis for each domain

The work presented offers a novel approach to the problem in the form of *heuristic estimation*, as well as several new methods of refining valid goal facts. Perhaps most importantly it offers a viable solution to the problem of offline library construction and allows **any** domain to be recognised without prior analysis.

AUTOGRAPH is **complete** in the sense that it can construct a hypothesis from *any* conjunction of literals within $\mathcal{H}$. The system is **scalable** because it is based on a 1st order approximation of the true goal hypothesis, meaning that the hypothesis grows only linearly with the size of the grounded problem. Finally, it is **domain-independent** because it only relies on the use of a standard problem definition schema and use of generic heuristics and algorithms.

A drawback of the system is the inability to know if a hypothesis is valid, due to the problem of detecting all mutually-exclusive propositions neing NP-Hard. Future work will explore the *approximation* of mutex information by recording facts which never appear together during intermediate plan-states.

The current linear convergence rate of the recogniser is to be expected from the heuristic estimation process, but a faster convergence rate would obviously be preferable. One method of increasing convergence rates could be to rule out any facts which cannot be reached within n steps, where $n > |P|$. However, in order to do this the problem of plan-length estimation would need to be solved first, along with the detection of accurate goal-conjunctions. Additionally, automating the process of selecting initial probabilities for each partition and during updates on a domain-by-domain basis using a system such as Hutter *et al* (2007) would reveal the optimal set of values for generating fast and accurate hypotheses.

References

[2003] Blaylock, N., and Allen, J. F. 2003. Corpus-based, statistical goal recognition. In *Proc. Int. Joint Conf. AI*, 1303–1308.

[2006] Blaylock, N., and Allen, J. F. 2006. Fast hierarchical goal schema recognition. In *Proc. Nat. Conf. on AI (AAAI)*, 796–801.

[2001] Bonet, B., and Geffner, H. 2001. Planning as heuristic search. *J. AI Res.* 129(1-2):5–33.

[2001] Carberry, S. 2001. Techniques for plan recognition. *User Modeling and User-Adapted Interaction* 11(1-2):31–48.

[1998] Fox, M., and Long, D. 1998. The automatic inference of state invariants in TIM. *J. AI Res.* 9:367–421.

[2003] Fox, M., and Long, D. 2003. PDDL2.1: An extension to PDDL for expressing temporal planning domains. *J. AI Res.* 20:61–124.

[2009] Geib, C. W. 2009. Delaying commitment in plan recognition using combinatory categorial grammars. In *Proc. Int. Joint Conf. on AI*, 1702–1707.

[2005] Gorniak, P., and Roy, D. 2005. Probabilistic grounding of situated speech using plan recognition and reference resolution. In *Proc. 7th Int. Conf. on Multimodal Interfaces*, 138–143.

[2004] Helmert, M. 2004. A planning heuristic based on causal graph analysis. In *Proc. Int. Conf. on Automated Planning and Scheduling*, 161–170.

[2009] Helmert, M. 2009. Concise finite-domain representations for PDDL planning tasks. *Artif. Intell.* 173(5-6):503–535.

[2001] Hoffmann, J., and Nebel, B. 2001. The FF planning system: Fast plan generation through heuristic search. *J. AI Res.* 14:253–302.

[2001] Hong, J. 2001. Goal recognition through goal graph analysis. *J. AI Res.* 15:1–30.

[2008] Huntemann, A.; Demeester, E.; van Brussel, H.; and Nuttin, M. 2008. Can Bayes help disabled users? A Bayesian approach to plan recognition and shared control for steering an electrical wheelchair. In *Proc. ACM/IEEE Human-Robot Interaction Conf.*, 67–70.

[2007] Hutter, F.; Babic, D.; Hoos, H. H.; and Hu, A. J. 2007. Boosting verification by automatic tuning of decision procedures. In *FMCAD '07: Proc. Formal Methods in CAD*, 27–34.

[1987] Kautz, H. A. 1987. *A formal theory of plan recognition*. Ph.D. Dissertation, University of Rochester.

[1995] Lesh, N., and Etzioni, O. 1995. A sound and fast goal recognizer. In *Proc. Int. Joint Conf. on AI*, 1704–1710.

[1997] Lesh, N. 1997. Adaptive goal recognition. In *Proc. Int. Joint Conf. on AI*, 1208–1214.

[2006] Mott, B.; Lee, S.; and Lester, J. 2006. Probabilistic goal recognition in interactive narrative environments. In *Proc. Nat. Conf. on AI (AAAI)*, 187–192.

[2009] Ramírez, M., and Geffner, H. 2009. Plan recognition as planning. In *Proc. Int. Joint Conf. on AI*, 1778–1783.

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