Chapter: *Plan Recognition in Hierarchical Planning*

Daniel Höller

Institute of Artificial Intelligence,
Ulm University, Germany

Winter Term 2018/2019
(Compiled on: February 20, 2019)

(Based on slides by Héctor Geffner)

ulm university universität uulm

Overview:

So far: Automated Planning

Given:

- Model of behavior.
  - What characterizes the world (with respect to a task)?
  - How can we/an agent change the world?
- Initial state – How does the world look like in the current situation?
- Goal and initial task definition – Which properties of the world do we want to achieve (classical planning) and how should the plans look like (hierarchical planning)?

Task:

- Find a sequence of actions that transforms the world from the initial state to a state that has the desired goal properties.

Plan and Goal Recognition

Given:

- Model of behavior.
  - What characterizes the world (with respect to a task)?
  - How can an agent change the world?
  - ***What are desirable goals?***
- Initial state – How does the world look like in the current situation?
- ***Observations*** – A sequence of actions some agent(s) has/have executed.

Task:

- Determine which of the goals the agent is pursuing.
- Determine what the agent is doing next.

Application Scenarios

- Robotics/software systems that interact with:
  - Humans and/or
  - other (autonomous) agents.

Application Scenarios

- Robotics/software systems that interact with:
  - Humans and/or
  - other (autonomous) agents.
- Intrusion detection:
  - Classify behavior.
  - Detect unusual behavior.

What are we doing?

Automated planning:

- Based on an abstraction of the world, we generate a plan to reach some goal.
- When the model is sufficiently precise, generated solutions will transform the system as intended.

What are we doing?

Automated planning:

- Based on an abstraction of the world, we generate a plan to reach some goal.
- When the model is sufficiently precise, generated solutions will transform the system as intended.

Plan and goal recognition:

- What must be fulfilled to classify some activity (e.g. some movement) as intentional action?

What are we doing?

Automated planning:

- Based on an abstraction of the world, we generate a plan to reach some goal.
- When the model is sufficiently precise, generated solutions will transform the system as intended.

Plan and goal recognition:

- What must be fulfilled to classify some activity (e.g. some movement) as intentional action?
- We make assumptions about reasons of behavior (the agent *wants to realize* something).

What are we doing?

Automated planning:

- Based on an abstraction of the world, we generate a plan to reach some goal.
- When the model is sufficiently precise, generated solutions will transform the system as intended.

Plan and goal recognition:

- What must be fulfilled to classify some activity (e.g. some movement) as intentional action?
- We make assumptions about reasons of behavior (the agent *wants to realize* something).
- We presume objectives of an agent and judge the helpfulness of actions regarding the objectives.

So, What *are* we Doing?

- We solve a combinatorial problem.
  - Decision problem: Is there a plan in line with the observations (this is a generalization of planning).

So, What *are* we Doing?

- We solve a combinatorial problem.
  - Decision problem: Is there a plan in line with the observations (this is a generalization of planning).
  - Practical: Find a plan that is in line with observations (and output the corresponding goal).

So, What *are* we Doing?

- We solve a combinatorial problem.
  - Decision problem: Is there a plan in line with the observations (this is a generalization of planning).
  - Practical: Find a plan that is in line with observations (and output the corresponding goal).
  - More practical: Return a probability distribution over possible goals/plans.

Assumptions

Agent

- Is the agent behaving rational?
- What does this mean?
  - Is *purchasing* a goal.
  - Is purchasing a goal *optimally*.

Assumptions

Agent

- Is the agent behaving rational?
- What does this mean?
  - Is *purchasing* a goal.
  - Is purchasing a goal *optimally*.
- Often assumed: “keyhole” observations (no contact to agent).

Assumptions

Agent

- Is the agent behaving rational?
- What does this mean?
  - Is *purchasing* a goal.
  - Is purchasing a goal *optimally*.
- Often assumed: “keyhole” observations (no contact to agent).
- Problematic: what is if the agent wants to confuse the observer (adversarial behavior, intrusion detection)?

Assumptions

Agent

- Is the agent behaving rational?
- What does this mean?
  - Is *purchasing* a goal.
  - Is purchasing a goal *optimally*.
- Often assumed: “keyhole” observations (no contact to agent).
- Problematic: what is if the agent wants to confuse the observer (adversarial behavior, intrusion detection)?
- Integration of PGR and Planning: cooperative behavior.

Assumptions

Agent

- Is the agent behaving rational?
- What does this mean?
  - Is *purchasing* a goal.
  - Is purchasing a goal *optimally*.
- Often assumed: “keyhole” observations (no contact to agent).
- Problematic: what is if the agent wants to confuse the observer (adversarial behavior, intrusion detection)?
- Integration of PGR and Planning: cooperative behavior.
- Is the model correct/complete?

Assumptions

Environment

- What is the input for the PGR system?

Assumptions

Environment

- What is the input for the PGR system?
  - Usually: actions ($\rightarrow$ Activity Recognition).
  - Sometimes: changes in the environment.

Assumptions

Environment

- What is the input for the PGR system?
  - Usually: actions ($\rightarrow$ Activity Recognition).
  - Sometimes: changes in the environment.
- Does the observer see all actions (missed observations)?

Assumptions

Environment

- What is the input for the PGR system?
  - Usually: actions ($\rightarrow$ Activity Recognition).
  - Sometimes: changes in the environment.
- Does the observer see all actions (missed observations)?
- Can we be wrong (False observations)?

Assumptions

Environment

- What is the input for the PGR system?
  - Usually: actions ($\rightarrow$ Activity Recognition).
  - Sometimes: changes in the environment.
- Does the observer see all actions (missed observations)?
- Can we be wrong (False observations)?

Goals

- Single/multiple goal?

Assumptions

Environment

- What is the input for the PGR system?
  - Usually: actions ($\rightarrow$ Activity Recognition).
  - Sometimes: changes in the environment.
- Does the observer see all actions (missed observations)?
- Can we be wrong (False observations)?

Goals

- Single/multiple goal?
- Plans interleaved?

Assumptions

Environment

- What is the input for the PGR system?
  - Usually: actions ($\rightarrow$ Activity Recognition).
  - Sometimes: changes in the environment.
- Does the observer see all actions (missed observations)?
- Can we be wrong (False observations)?

Goals

- Single/multiple goal?
- Plans interleaved?
- Static/dynamic?

Approaches

Based on

- plan libraries
- Parsing
- Probabilistic models
- ...
- Most recent approach: Plan Recognition as Planning

Plan Recognition as Planning

- Benefit from research in planning.

Plan Recognition as Planning

- Benefit from research in planning.
- Well-established formalisms.

Plan Recognition as Planning

- Benefit from research in planning.
- Well-established formalisms.
- Use the efficient solvers that are available in planning.

Plan Recognition as Planning

- Benefit from research in planning.
- Well-established formalisms.
- Use the efficient solvers that are available in planning.
- We will have a look at tree approaches:

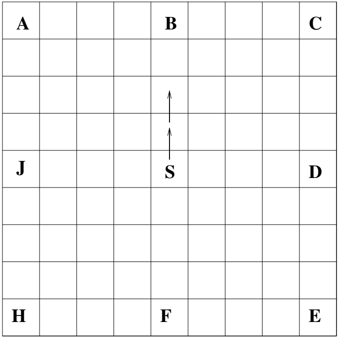
Plan Recognition as Planning

- Benefit from research in planning.
- Well-established formalisms.
- Use the efficient solvers that are available in planning.
- We will have a look at tree approaches:
  - Next: Two approaches based on STRIPS (slides by Héctor Geffner).

Plan Recognition as Planning

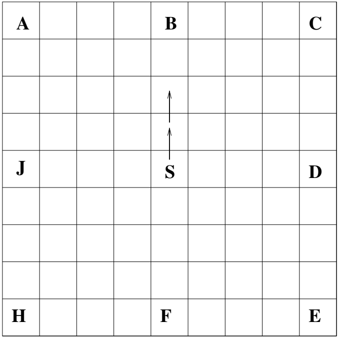
- Benefit from research in planning.
- Well-established formalisms.
- Use the efficient solvers that are available in planning.
- We will have a look at tree approaches:
  - Next: Two approaches based on STRIPS (slides by Héctor Geffner).
  - Afterwards: Based on Hierarchical Task Network (HTN) planning.

Plan Recognition



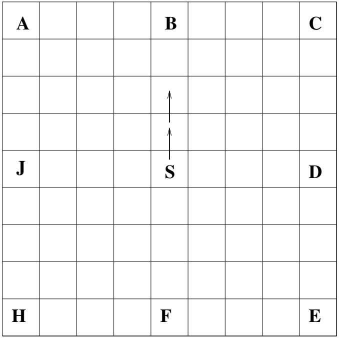
- Agent can move one unit in the four directions.

Plan Recognition



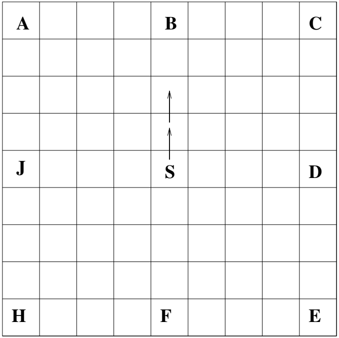
- Agent can move one unit in the four directions.
- Possible targets are A, B, C, ...

Plan Recognition



- Agent can move one unit in the four directions.
- Possible targets are A, B, C, . . .
- Starting in S, he is observed to move up twice.

Plan Recognition



- Agent can move one unit in the four directions.
- Possible targets are A, B, C, . . .
- Starting in S, he is observed to move up twice.
- Where is he going?

Standard Plan Recognition over Libraries (Abstract View)

- A plan recognition problem defined by triple $T = (\mathcal{G}, \Pi, O)$, where
  - $\mathcal{G}$ is the set of possible goals G.
  - $\Pi(G)$ is the set of possible plans π for G, $G \subseteq \mathcal{G}$.
  - O is an observation sequence $a_1, \ldots a_n$ where each a_i is an action.

Standard Plan Recognition over Libraries (Abstract View)

- A plan recognition problem defined by triple $T = (\mathcal{G}, \Pi, O)$, where
  - $\mathcal{G}$ is the set of possible goals G.
  - $\Pi(G)$ is the set of possible plans π for G, $G \subseteq \mathcal{G}$.
  - O is an observation sequence $a_1, \ldots a_n$ where each a_i is an action.
- A possible goal $G \in \mathcal{G}$ is plausible if $\exists$ plan π in Π(G) that satisfies O.
- An action sequence π satisfies O if O is a subsequence of π.

(Classical) Plan Recognition over Action Theories

- PR over classical planning domains is similar but with set of plans Π(G) defined implicitly:
- A plan recognition problem is a triplet $T = (P, \mathcal{G}, O)$, where
  - $P = (F, A, I)$ is planning domain: fluents F, actions A, init I, no goal.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots a_n$, with $a_i \in A$.

(Classical) Plan Recognition over Action Theories

- PR over classical planning domains is similar but with set of plans Π(G) defined implicitly:
- A plan recognition problem is a triplet $T = (P, \mathcal{G}, O)$, where
  - $P = (F, A, I)$ is planning domain: fluents F, actions A, init I, no goal.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots a_n$, with $a_i \in A$.
- If $\Pi(G)$ stands for "good plans" for G in P (to be defined), then as before:
  - A possible goal $G \in \mathcal{G}$ is plausible if there is a plan π in $\Pi(G)$ that satisfies O.
  - An action sequence π satisfies O if O is a subsequence of π.

(Classical) Plan Recognition over Action Theories

- PR over classical planning domains is similar but with set of plans Π(G) defined implicitly:
- A plan recognition problem is a triplet $T = (P, \mathcal{G}, O)$, where
  - $P = (F, A, I)$ is planning domain: fluents F, actions A, init I, no goal.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots a_n$, with $a_i \in A$.
- If $\Pi(G)$ stands for "good plans" for G in P (to be defined), then as before:
  - A possible goal $G \in \mathcal{G}$ is plausible if there is a plan π in $\Pi(G)$ that satisfies O.
  - An action sequence π satisfies O if O is a subsequence of π.
- Our goal: define the good plans and solve the problem with a classical planner.

Plan Recognition as Planning: First Formulation

Define the set $\Pi(G)$ of "good plans" for G in P, as the optimal plans for G in P.

- Then $G \in \mathcal{G}$ is a plausible goal given observations O:
  - Iff there is an optimal plan π for G in P that satisfies O;
  - iff there is an optimal plan π for G in P that is a plan for $G + O$ in P';
  - iff cost of G in P equal to cost of $G + O$ in P' abbreviated

$$c'_P(G + O) = c_P(G)$$

Plan Recognition as Planning: First Formulation

Define the set $\Pi(G)$ of "good plans" for G in P, as the optimal plans for G in P.

- Then $G \in \mathcal{G}$ is a plausible goal given observations O:
  - Iff there is an optimal plan π for G in P that satisfies O;
  - iff there is an optimal plan π for G in P that is a plan for $G + O$ in P';
  - iff cost of G in P equal to cost of $G + O$ in P' abbreviated

$$c'_P(G + O) = c_P(G)$$

- Given that we can create a "new" planning problem solving $G + O$,
- it follows that plausibility of G can be computed exactly by calling an optimal planner twice: one for computing $c'_P(G + O)$, one for computing $c_P(G)$.

Compiling Observations Away

We get rid of observations O by transforming $P = (F, I, A)$ into $P' = (F', I', A')$ so that

- π is a plan for G in P that satisfies O iff π is a plan for $G + O$ in P'.

and

- π is a plan for G in P that doesn't satisfy O iff π is a plan for $G + \overline{O}$ in P'.

The transformation from P into P' is quite simple.

Compiling Observations Away

- Given $P = (F, I, A)$, the transformed problem is $P' = (F', I', A')$:
  - $F' = F \cup \{p_a \mid a \in O\}$, where p_a is new fluent for the observed action a.
  - $I' = I$.
  - $A' = A$.

Compiling Observations Away

- Given $P = (F, I, A)$, the transformed problem is $P' = (F', I', A')$:
  - $F' = F \cup \{p_a \mid a \in O\}$, where p_a is new fluent for the observed action a.
  - $I' = I$.
  - $A' = A$.
- The actions $a \in O$ have an extra effect in A':
  - p_a, if a is the first observation in O, and
  - $p_b \rightarrow p_a$, if b is the action that immediately precedes a in O.

Compiling Observations Away

- Given $P = (F, I, A)$, the transformed problem is $P' = (F', I', A')$:
  - $F' = F \cup \{p_a \mid a \in O\}$, where p_a is new fluent for the observed action a.
  - $I' = I$.
  - $A' = A$.
- The actions $a \in O$ have an extra effect in A':
  - p_a, if a is the first observation in O, and
  - $p_b \rightarrow p_a$, if b is the action that immediately precedes a in O.
- The “goals” O and $\overline{O}$ in P' are p_a and $\neg p_a$ for the last action a in O.
- The plans π for G in P that satisfy/don’t satisfy O are the plans in P' for $G + O / G + \overline{O}$ respectively.

Plan Recognition as Planning: A More General Formulation

- So far we filter goals G as plausible or implausible.

Plan Recognition as Planning: A More General Formulation

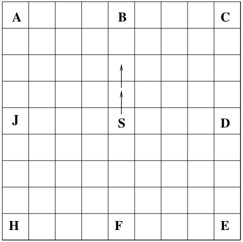
- So far we filter goals G as plausible or implausible.
- Rather rank them with a probability distribution $P(G|O)$, $G \in \mathcal{G}$.

Plan Recognition as Planning: A More General Formulation

- So far we filter goals G as plausible or implausible.
- Rather rank them with a probability distribution $P(G|O)$, $G \in \mathcal{G}$.
- From Bayes Rule $P(G|O) = \alpha P(O|G)P(G)$, where
  - α is a normalizing constant.
  - $P(G)$ is assumed to be given in problem specification.
  - $P(O|G)$ is defined in terms of extra cost to pay for not complying with the observations O:

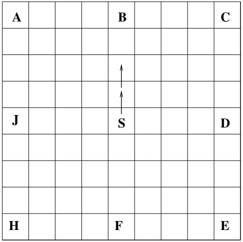
$$P(O|G) = \mathit{function}(c(G + \overline{O}) - c(G + O))$$

Example: Navigation in a Grid Revisited



If $\Delta(G, O)$ defined as $c(G + \overline{O}) - c(G + O)$:

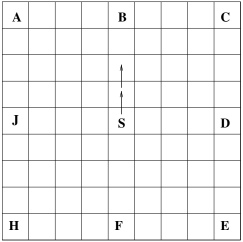
Example: Navigation in a Grid Revisited



If $\Delta(G, O)$ defined as $c(G + \overline{O}) - c(G + O)$:

- For $G = B$, $c(B + O) = c(B) = 4$; $c(B + \overline{O}) = 6$; thus $\Delta(B, O) = 2$.

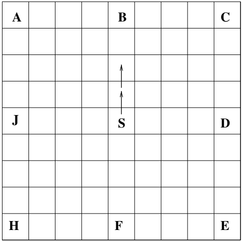
Example: Navigation in a Grid Revisited



If $\Delta(G, O)$ defined as $c(G + \overline{O}) - c(G + O)$:

- For $G = B$, $c(B + O) = c(B) = 4$; $c(B + \overline{O}) = 6$; thus $\Delta(B, O) = 2$.
- For $G = C$ or A, $c(C + O) = c(C + \overline{O}) = c(C) = 8$; thus $\Delta(C, O) = 0$.

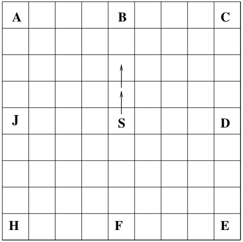
Example: Navigation in a Grid Revisited



If $\Delta(G, O)$ defined as $c(G + \overline{O}) - c(G + O)$:

- For $G = B$, $c(B + O) = c(B) = 4$; $c(B + \overline{O}) = 6$; thus $\Delta(B, O) = 2$.
- For $G = C$ or A, $c(C + O) = c(C + \overline{O}) = c(C) = 8$; thus $\Delta(C, O) = 0$.
- For all other G, $c(G + O) = 8$; $c(G + \overline{O}) = c(G) = 4$; thus $\Delta(G, O) = -4$.

Example: Navigation in a Grid Revisited



If $\Delta(G, O)$ defined as $c(G + \overline{O}) - c(G + O)$:

- For $G = B$, $c(B + O) = c(B) = 4$; $c(B + \overline{O}) = 6$; thus $\Delta(B, O) = 2$.
- For $G = C$ or A, $c(C + O) = c(C + \overline{O}) = c(C) = 8$; thus $\Delta(C, O) = 0$.
- For all other G, $c(G + O) = 8$; $c(G + \overline{O}) = c(G) = 4$; thus $\Delta(G, O) = -4$.

If $P(O|G)$ is a monotonic function of $\Delta(G, O)$, then

$$P(O|B) > [P(O|C) = P(O|A)] > P(O|G), \text{ for } G \notin \{A, B, C\}.$$

Defining the Likelihoods $P(O|G)$

- Assuming Boltzmann distribution and writing $exp\{x\}$ for e^x, likelihoods become

$$P(O|G) = \alpha exp\{-\beta c(G + O)\}$$
$$P(\overline{O}|G) = \alpha exp\{-\beta c(G + \overline{O})\}$$

where α is a normalizing constant, and β is a positive constant.

Defining the Likelihoods $P(O|G)$

- Assuming Boltzmann distribution and writing $exp\{x\}$ for e^x, likelihoods become

$$P(O|G) = \alpha exp\{-\beta c(G + O)\}$$
$$P(\overline{O}|G) = \alpha exp\{-\beta c(G + \overline{O})\}$$

  where α is a normalizing constant, and β is a positive constant.

- Taking ratio of two equations, it follows that

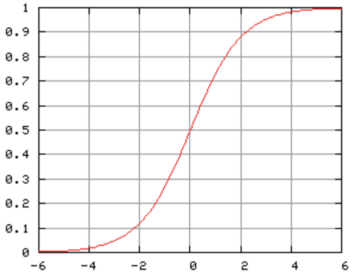
$$P(O|G)/P(\overline{O}|G) = exp\{\beta\Delta(G, O)\}$$

  and hence

$$P(O|G) = 1/(1 + exp\{-\beta\Delta(G, O)\}) = sigmoid(\beta\Delta(G, O))$$

(whiteboard)

Defining the Likelihoods $P(O|G)$



$$P(O|G) = sigmoid(\beta\Delta(G,O))$$
$$\Delta(G,O) = c(G+\overline{O}) - c(G+O)$$

E.g.,

$$P(O|G) < P(\overline{O}|G) \text{ if } c(G+\overline{O}) < c(G+O)$$
$$P(O|G) = 1 \text{ if } c(G+O) < c(G+\overline{O}) = \infty$$

Probabilistic Plan Recognition as Planning: Summary

- A plan recognition problem is a tuple $T = (P, \mathcal{G}, O, Prob)$ where
  - P is a planning domain $P = (F, I, A)$.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots, a_n$, $a_i \in O$.
  - $Prob$ is prior distribution over $\mathcal{G}$.

Probabilistic Plan Recognition as Planning: Summary

- A plan recognition problem is a tuple $T = (P, \mathcal{G}, O, Prob)$ where
  - P is a planning domain $P = (F, I, A)$.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots, a_n$, $a_i \in O$.
  - $Prob$ is prior distribution over $\mathcal{G}$.
- Posterior distribution $P(G|O)$ obtained from:
  - Bayes Rule $P(G|O) = \alpha P(O|G) Prob(G)$ and
  - Likelihood $P(O|G) = sigmoid\{\beta[c(G + \overline{O}) - c(G + O)]\}$.

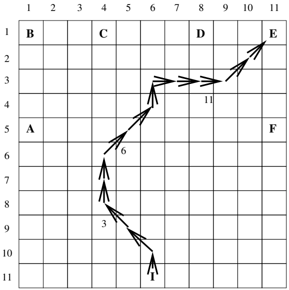
Probabilistic Plan Recognition as Planning: Summary

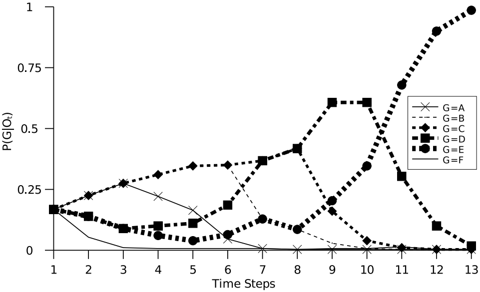
- A plan recognition problem is a tuple $T = (P, \mathcal{G}, O, Prob)$ where
  - P is a planning domain $P = (F, I, A)$.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots, a_n$, $a_i \in O$.
  - $Prob$ is prior distribution over $\mathcal{G}$.
- Posterior distribution $P(G|O)$ obtained from:
  - Bayes Rule $P(G|O) = \alpha P(O|G) Prob(G)$ and
  - Likelihood $P(O|G) = sigmoid\{\beta[c(G + \overline{O}) - c(G + O)]\}$.
- Distribution $P(G|O)$ computed exactly or approximately:
  - exactly using optimal planner for determining $c(G + O)$ and $c(G + \overline{O})$ and
  - approximately using suboptimal planner for $c(G + O)$ and $c(G + \overline{O})$.

Probabilistic Plan Recognition as Planning: Summary

- A plan recognition problem is a tuple $T = (P, \mathcal{G}, O, Prob)$ where
  - P is a planning domain $P = (F, I, A)$.
  - $\mathcal{G}$ is a set of possible goals G, $G \subseteq F$.
  - O is the observation sequence $a_1, \ldots, a_n$, $a_i \in O$.
  - $Prob$ is prior distribution over $\mathcal{G}$.
- Posterior distribution $P(G|O)$ obtained from:
  - Bayes Rule $P(G|O) = \alpha P(O|G) Prob(G)$ and
  - Likelihood $P(O|G) = sigmoid\{\beta[c(G + \overline{O}) - c(G + O)]\}$.
- Distribution $P(G|O)$ computed exactly or approximately:
  - exactly using optimal planner for determining $c(G + O)$ and $c(G + \overline{O})$ and
  - approximately using suboptimal planner for $c(G + O)$ and $c(G + \overline{O})$.
- In either case, $2 \times |G|$ planner calls are needed.

Example: Noisy Walk





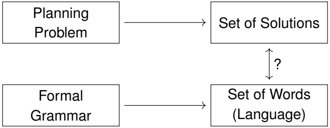
Graph on the left shows “noisy walk” and possible targets; curves on the right show posterior $P(G|O)$ of each possible target G as a function of time.

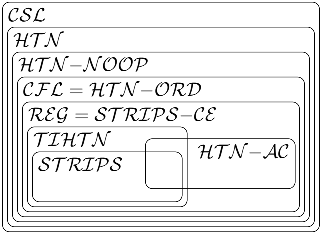
Plan and Goal Recognition as HTN Planning

- STRIPS: specifies what has to be in a plan.
- HTN: also excludes other elements from being in the plan.
  - No actions apart from hierarchy.
  - Enables to plan only once, regardless how many goals there are.
  - Interesting to rule out non-fitting plans/goals.

Plan and Goal Recognition as HTN Planning

- STRIPS: specifies what has to be in a plan.
- HTN: also excludes other elements from being in the plan.
  - No actions apart from hierarchy.
  - Enables to plan only once, regardless how many goals there are.
  - Interesting to rule out non-fitting plans/goals.
- Formalism much more expressive than STRIPS planning.





Problem Definition

Let $\mathcal{D} = (V, P, \delta, C, M)$ be an HTN planning domain.

Definition (Plan and Goal Recognition Problem)

A *PGR problem* $(\mathcal{D}, s_I, O, \mathcal{G})$ extends the model by:

- s_0 the initial state.
- $O = \langle o_1, o_2, \ldots, o_m \rangle$ the sequence of observations.
- $\mathcal{G} = \{G_1, G_2, \ldots, G_r\}$ a set of possible *goal (task) networks*.

Problem Definition

Let $\mathcal{D} = (V, P, \delta, C, M)$ be an HTN planning domain.

Definition (Plan and Goal Recognition Problem)

A *PGR problem* $(\mathcal{D}, s_I, O, \mathcal{G})$ extends the model by:

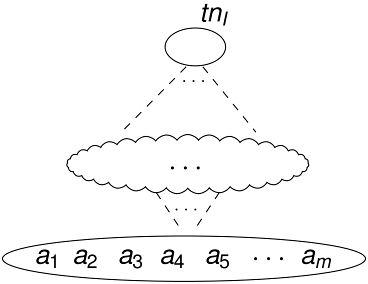
- s_0 the initial state.
- $O = \langle o_1, o_2, \ldots, o_m \rangle$ the sequence of observations.
- $\mathcal{G} = \{G_1, G_2, \ldots, G_r\}$ a set of possible *goal (task) networks*.

Definition (Recognized Plan and Goal)

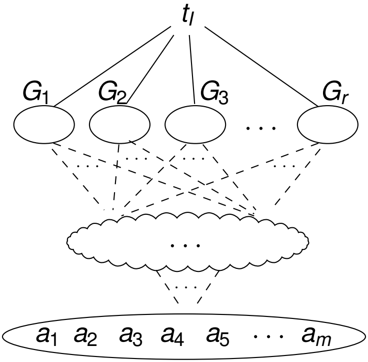
Given a PGR problem, a *goal* $G_i \in \mathcal{G}$ *explains* the observations $\langle o_1, o_2, \ldots, o_m \rangle$ iff there is a solution $s \in Sol(\mathcal{D}, s_0, G_i)$ with an executable linearization $\langle a_1, a_2, \ldots, a_n \rangle$ of its tasks with:

- $n \geq m$ and $o_i = a_i$ for $1 \leq i \leq m$.
- $\langle a_1, a_2, \ldots, a_n \rangle$ is the recognized *plan*.

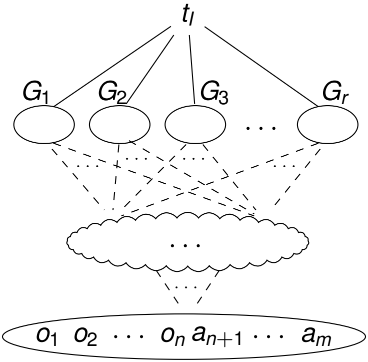
Approach



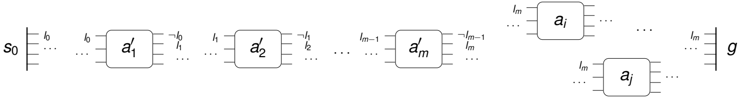
Approach – Make Goals Reachable



Approach – Enforce a Prefix



Approach – Enforce a Prefix

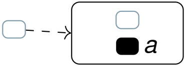


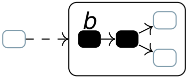
- Copy observed actions: $a \rightarrow a, a'$
- a' gets new preconditions and effects, it can only be placed at the position where it has been observed
- a is modified such that they can only be placed after the prefix
- Observed actions have to be in the plan

Approach – Enforce a Prefix

Observations: *ab*

Set of methods:



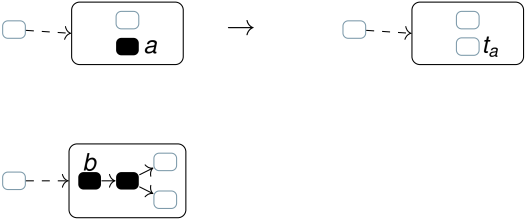


...

Approach – Enforce a Prefix

Observations: *ab*

Set of methods:

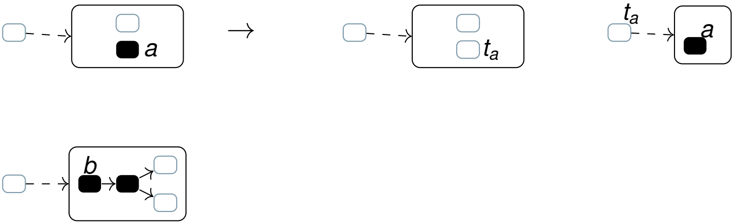


. . .

Approach – Enforce a Prefix

Observations: *ab*

Set of methods:

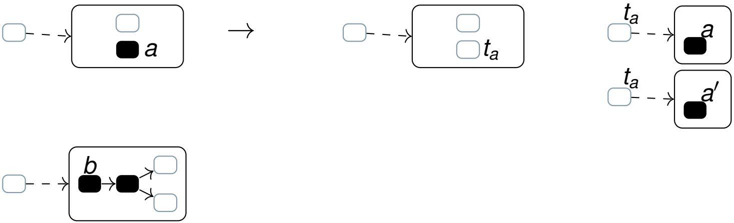


...

Approach – Enforce a Prefix

Observations: *ab*

Set of methods:

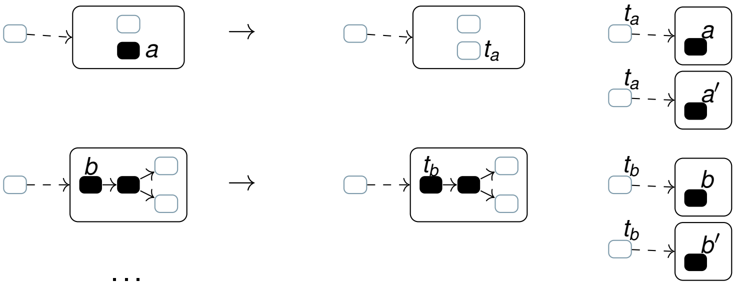


...

Approach – Enforce a Prefix

Observations: *ab*

Set of methods:



Approach

- Off-the-shelf HTN planners solve the PGR problem.
- Needing a single run of the planner, regardless how many goals there are.

Approach

- Off-the-shelf HTN planners solve the PGR problem.
- Needing a single run of the planner, regardless how many goals there are.

Example: Simple cooking domain.

- There is a short plan for making noodles.
- There is a long plan for making tiramisu (needing eggs).
- Observation: an egg was broken.

Approach

- Off-the-shelf HTN planners solve the PGR problem.
- Needing a single run of the planner, regardless how many goals there are.

Example: Simple cooking domain.

- There is a short plan for making noodles.
- There is a long plan for making tiramisu (needing eggs).
- Observation: an egg was broken.

HTN planning behavior:

- HTN planner is forced to generate the long plan.
- Planner will tend to find a short/cheap plan.

Approach

- Off-the-shelf HTN planners solve the PGR problem.
- Needing a single run of the planner, regardless how many goals there are.

Example: Simple cooking domain.

- There is a short plan for making noodles.
- There is a long plan for making tiramisu (needing eggs).
- Observation: an egg was broken.

HTN planning behavior:

- HTN planner is forced to generate the long plan.
- Planner will tend to find a short/cheap plan.

Consider a similar (goal) encoding in STRIPS:

- Planner might start with “open egg” and make noodles afterwards.
- This might even be cheaper when planning optimally.

Plan and Goal Recognition (as Planning)

- We studied the *Plan and Goal Recognition* problem.
- Assumptions are more problematic than in planning.

Plan and Goal Recognition (as Planning)

- We studied the *Plan and Goal Recognition* problem.
- Assumptions are more problematic than in planning.
- Less “converged” problem definition:
  - type of behavior model,
  - what is an observation,
  - what is computed,
  - . . .

Plan and Goal Recognition (as Planning)

- We studied the *Plan and Goal Recognition* problem.
- Assumptions are more problematic than in planning.
- Less "converged" problem definition:
  - type of behavior model,
  - what is an observation,
  - what is computed,
  - . . .
- One way to solve it: "Plan Recognition as Planning".
- Compilation to planning, we have seen:
  - 1st transformation into classical planning:
    Miquel Ramírez and Héctor Geffner. "Plan Recognition as Planning". In: *Proc. of the 21st Int. Joint Conf. on Artificial Intelligence (IJCAI 2009)*. AAAI Press, 2009, pp. 1778–1783
  - 2nd transformation, enabling the computation of probability distributions:
    Miquel Ramírez and Hector Geffner. "Probabilistic Plan Recognition Using Off-the-Shelf Classical Planners". In: *Proc. of the 24th AAAI Conf. on Artificial Intelligence (AAAI 2010)*. AAAI Press, 2010
  - Transformation into the more expressive HTN formalism.