LP-Based Approaches for Goal Recognition as Planning

Felipe Meneguzzi†
André Grahl Pereira
Ramon Fraga Pereira

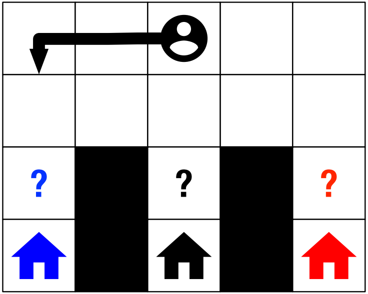
†Pontifical Catholic University of Rio Grande do Sul, Brazil
Seconded at the University of Melbourne
felipe.meneguzzi@pucrs.br

Melbourne, February, 2020

Table of Contents

What is it?

- **Goal Recognition** is the task of recognizing agents' goal that explains a sequence of observations of its actions;
  - Related to plan recognition, i.e. recognizing a *top-level* action
  - A specific form of the problem of abduction
- Roughly two types of approach:
  - Plan-library based (*classical* plan recognition)
  - Domain-theory based (plan recognition as planning, or PRAP)



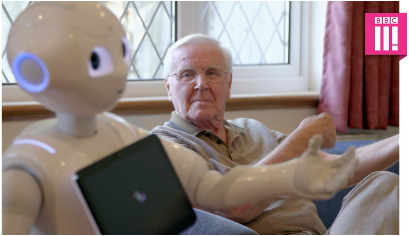
Why do we need goal recognition?

- Recognizing plans and goals of others is critical for meaningful interaction:
  - important for humans/agents working in the same environment
  - increasingly important as we build more intelligent systems

- Overall area of Plan, Activity and Intent Recognition
  - Activity recognition: recognizing meaningful activities from low-level sensor data
  - Plan/Intent/Goal recognition: recognizing intentional higher-level sequences of activities

Why do we need goal recognition?

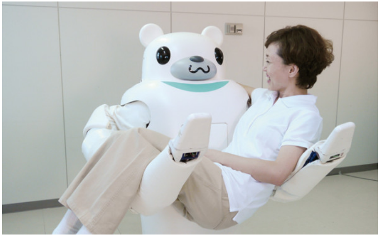
- Recognizing plans and goals of others is critical for meaningful interaction:
  - important for humans/agents working in the same environment
  - increasingly important as we build more intelligent systems



- Overall area of Plan, Activity and Intent Recognition
  - Activity recognition: recognizing meaningful activities from low-level sensor data
  - Plan/Intent/Goal recognition: recognizing intentional higher-level sequences of activities

Why do we need goal recognition?

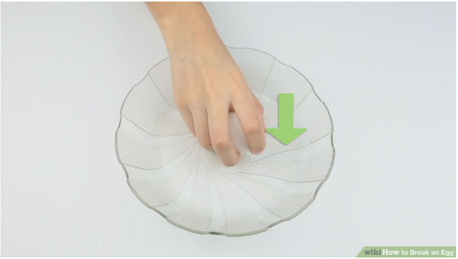
- Recognizing plans and goals of others is critical for meaningful interaction:
  - important for humans/agents working in the same environment
  - increasingly important as we build more intelligent systems



- Overall area of Plan, Activity and Intent Recognition
  - Activity recognition: recognizing meaningful activities from low-level sensor data
  - Plan/Intent/Goal recognition: recognizing intentional higher-level sequences of activities

An example of Activity Recognition

An example of Activity Recognition



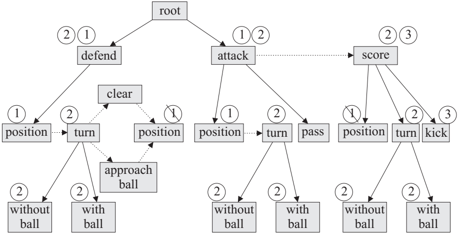
An example of Activity Recognition

An example of Activity Recognition

breaking egg

Flavors of Recognition Formalism

Plan Library



Domain Theory (PRAP)

```
(define (domain grid)
(:requirements :strips :typing)
(:types place shape key)
(:predicates (conn ?x ?y - place)
             (key-shape ?k - key ?s - shape)
             (lock-shape ?x - place ?s - shape)
             (at ?r - key ?x - place )
             (at-robot ?x - place)
             (locked ?x - place)
             (carrying ?k - key)
             (open ?x - place)
)

(:action unlock
:parameters (?curpos ?lockpos - place ?key - key ?shape - shape)
:precondition (and (conn ?curpos ?lockpos) (key-shape ?key ?shape)
                   (lock-shape ?lockpos ?shape) (at-robot ?curpos)
                   (locked ?lockpos) (carrying ?key))
:effect (and  (open ?lockpos) (not (locked ?lockpos)))
)

(:action move
:parameters (?curpos ?nextpos - place)
:precondition (and (at-robot ?curpos) (conn ?curpos ?nextpos) (open ?n
:effect (and (at-robot ?nextpos) (not (at-robot ?curpos)))
)

(:action pickup
:parameters (?curpos - place ?key - key)
:precondition (and (at-robot ?curpos) (at ?key ?curpos))
:effect (and (carrying ?key)
   (not (at ?key ?curpos)))
)

)
```

Table of Contents

Automated Planning

Definition (**Planning**)

A planning instance is represented by a triple $\Pi = \langle \Xi, \mathcal{I}, G \rangle$, in which:

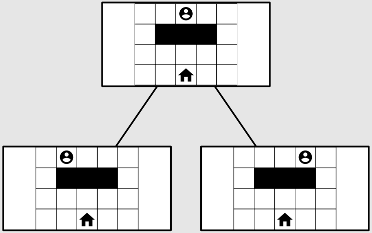
- $\Xi = \langle \Sigma, \mathcal{A} \rangle$ is the **domain definition**, and consists of a finite set of **facts** Σ and a finite set of **actions** $\mathcal{A}$ (action costs typically 1);
- $\mathcal{I} \subseteq \Sigma$ and $G \subseteq \Sigma$ represent the **planning problem**, in which $\mathcal{I} \subseteq \Sigma$ is the **initial state**, and $G \subseteq \Sigma$ is the **goal state**.

- Actions $a \in \mathcal{A}$ are tuples $a = \langle pre(a), eff(a), cost(a) \rangle$
- Facts Σ can be modeled in a variety of ways:
  - As a logic language (restricted FOL):
    states are truth assignments
  - As a set of variables $\mathcal{V}$ with finite domains:
    states are variable assignments

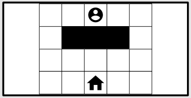
Automated Planning - Less boring

Planning problems have three key ingredients

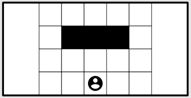
Domain Description



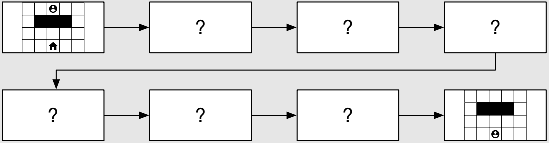
Initial State



Goal State



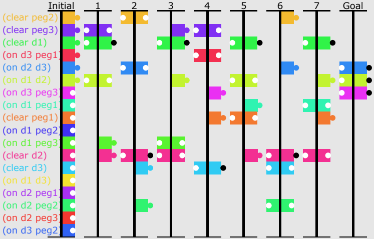
Solution



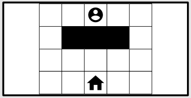
Automated Planning - Less boring

Planning problems have three key ingredients

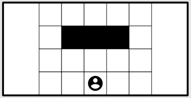
Domain Description



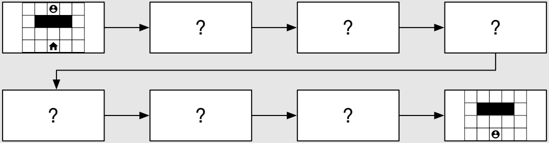
Initial State



Goal State



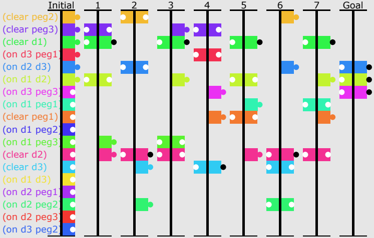
Solution



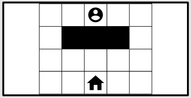
Automated Planning - Less boring

Planning problems have three key ingredients

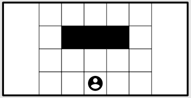
Domain Description



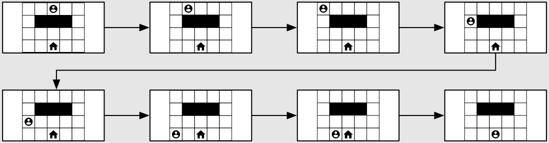
Initial State



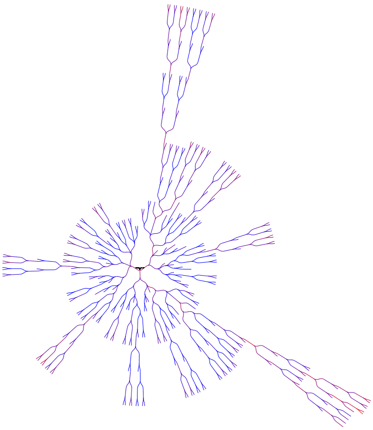
Goal State



Solution



Planning Heuristics

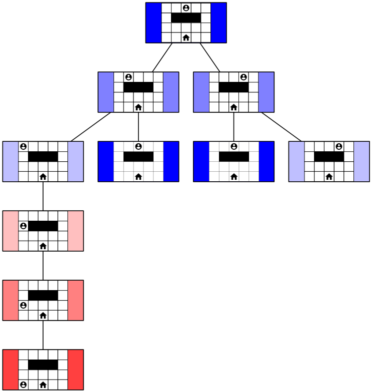


Most modern planners rely on heuristics to efficiently search the state-space.

Two key challenges in research on novel heuristics

- Informativeness of the heuristic
- Computational efficiency

Planning Heuristics



Most modern planners rely on heuristics to efficiently search the state-space.

Two key challenges in research on novel heuristics

- Informativeness of the heuristic
- Computational efficiency

Goal Recognition Problem

Definition (Goal Recognition Problem)

A goal recognition problem is a tuple $P = \langle \Xi, \mathcal{I}, \mathcal{G}, \mathrm{O} \rangle$, where:

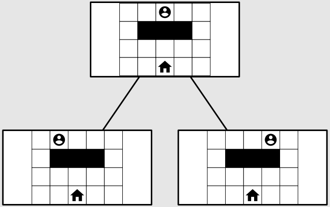
- $\Xi = \langle \Sigma, \mathcal{A} \rangle$ is the domain definition (facts and actions) ;
- $\mathcal{I} \subseteq \Sigma$ is the initial state;
- $\mathcal{G}$ s.t. $\forall G \in \mathcal{G}, G \subseteq \Sigma$ is a set of candidate goals (with an assumed hidden goal G); and
- O is a sequence $\langle o_1, \ldots o_n \rangle$ of observations, where $o_i \in \mathcal{A}$

- The solution for a goal recognition problem is the hidden goal $G \in \mathcal{G}$ that is most consistent with observation sequence O.
- Caveat: we may have other representations for the observations
- This is what I will refer to as PRAP

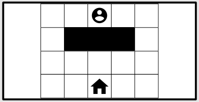
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have three key ingredients

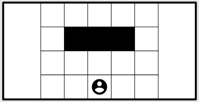
Domain Description



Initial State



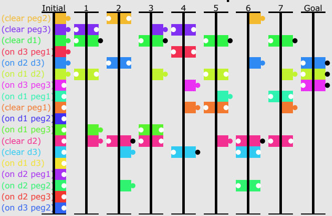
Goal State



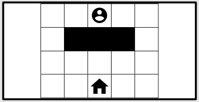
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

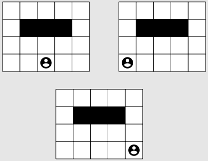
Domain Description



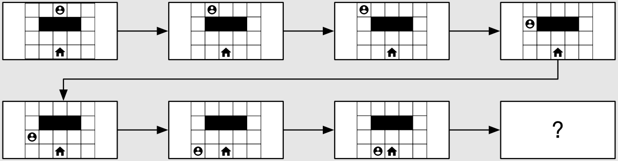
Initial State



Goal **Hypotheses**



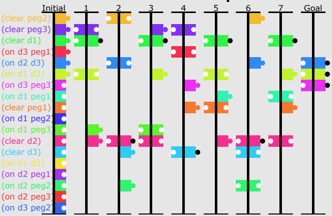
Observations



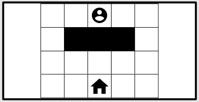
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

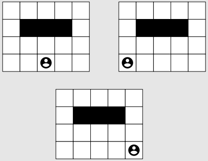
Domain Description



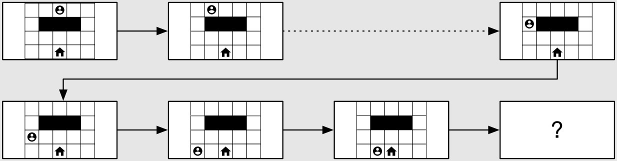
Initial State



Goal **Hypotheses**



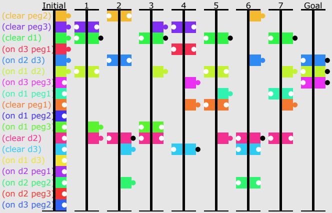
Observations



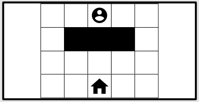
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

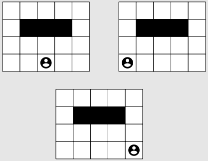
Domain Description



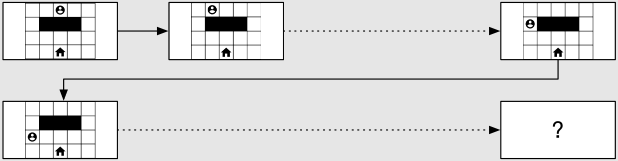
Initial State



Goal **Hypotheses**



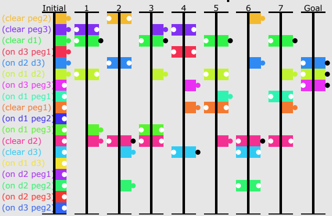
Observations



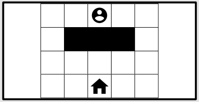
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

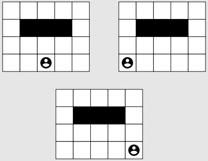
Domain Description



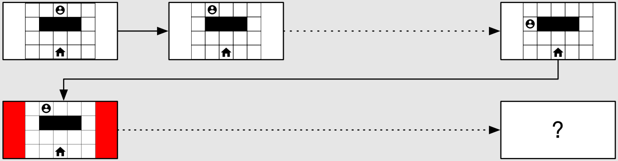
Initial State



Goal **Hypotheses**

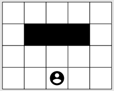


Observations



Solution

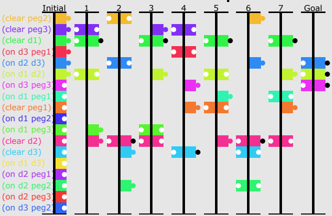
Correct Goal



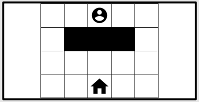
Goal Recognition Problem - Less boring

Goal/Plan Recognition problems have **four** key ingredients

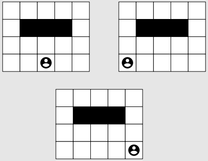
Domain Description



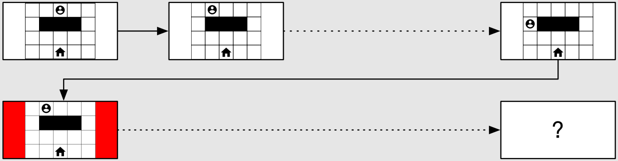
Initial State



Goal **Hypotheses**



Observations



Solution

Probability Distribution

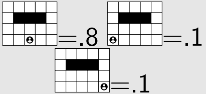


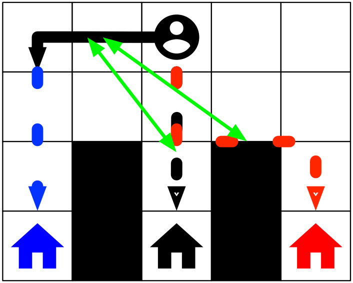
Table of Contents

Goal Recognition using Planning Domains I

Ramirez and Geffner (2009 and 2010)

- First approaches to goal recognition: Plan Recognition as Planning (PRAP)
- Probabilistic model aims to compute $P(G \mid O)$
- Following Bayes Rule $P(G \mid O) = \alpha P(O \mid G) P(G)$
- Given $P(G)$ as a prior, key bottleneck is computing $P(O \mid G)$

- Compute $P(O \mid G)$ in terms of a cost difference $c(G, O) - c(G, \bar{O})$
- Costs **two planner calls per goal hypothesis**

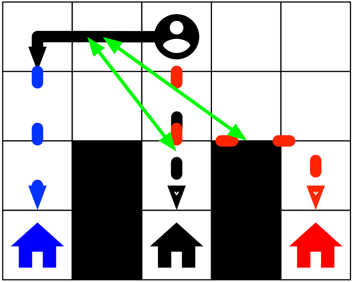


Goal Recognition using Planning Domains II

Sohrabi et al. (2016)

- Conceptually similar to Ramirez and Geffner: aims to compute $P(G \mid O)$ via $\alpha P(O \mid G)P(G)$
- Compilation of plan recognition problem into **multiple planning** problems (one for each G)

- Compute Top-k or diverse plans π to approximate $P(O \mid G) = \sum_{\pi} P(O \mid \pi) \cdot P(\pi \mid G)$
- Compensate noisy observations by imposing a cost on dropped Observations



Goal Recognition using Planning Heuristics

Pereira, Oren and Meneguzzi (2017):

- **Obviate the need to execute a planner multiple times** for recognizing goals; and
- Novel goal recognition heuristics that use **planning landmarks**.
- **More accurate** and **orders of magnitude faster** than all previous approaches.

Planning Landmarks:

- Are **necessary conditions** for any valid plan
- Theoretical cost of computation is the same as planning

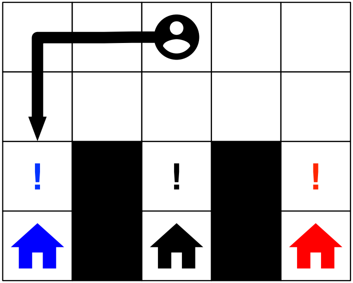


Table of Contents

Operator Counting Heuristics

- Based on the idea of *Cost Partitioning for Landmarks*
- Represents cost of a planning problem in terms of linear constraints:[1]
  - **Variables:** Count_o for each operator o
  - **Objective:** Minimize $\sum_{o} \text{Count}_o \cdot \text{cost}(o)$, subject to
    - $\sum_{o \in L} \text{Count}_o \geq 1$ for all landmarks L
    - $\text{Count}_o \geq 0$ for all operators o
  - Numbers of operator occurrences in any plan satisfy constraints
  - Minimizing total cost $\rightarrow$ admissible heuristic

[1] Adapted from Helmert and Röger's planning course

Operator Counting

Operator-counting Constraints[2]

- *linear constraints* whose variables denote *number of occurrences* of a given operator
- must be satisfied by every plan

Examples:

- $\text{Count}_{o1} + \text{Count}_{o2} \geq 1$ "must use o_1 or o_2 at least once"
- $\text{Count}_{o1} - \text{Count}_{o3} \leq 0$ "cannot use o_1 more often than o_3"

Motivation:

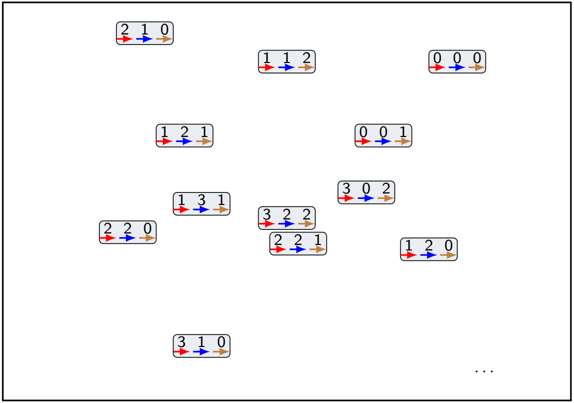
- declarative way to **represent knowledge** about the solution
- allows **reasoning about solutions** to derive heuristic estimates
- elegant framework to combine information from multiple heuristics

[2] Adapted from Helmert and Röger's planning course

Operator Counting heuristics

OC variables correspond to unordered actions in potential plans.[3]

Operator occurrences in potential plans

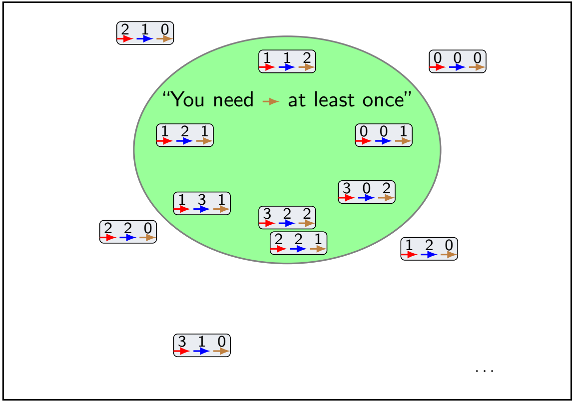


[3] Adapted from Helmert and Röger's planning course

Operator Counting heuristics

OC variables correspond to unordered actions in potential plans.[3]

Operator occurrences in potential plans

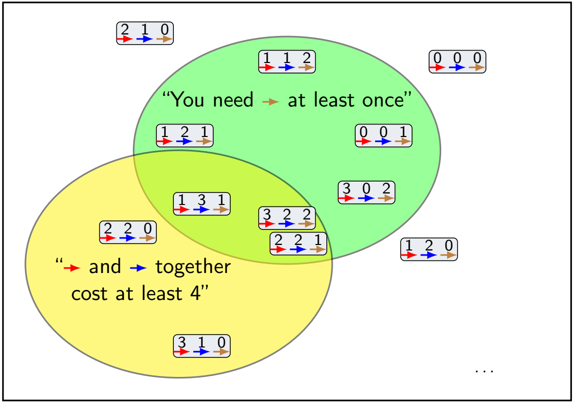


[3] Adapted from Helmert and Röger's planning course

Operator Counting heuristics

OC variables correspond to unordered actions in potential plans.[3]

Operator occurrences in potential plans

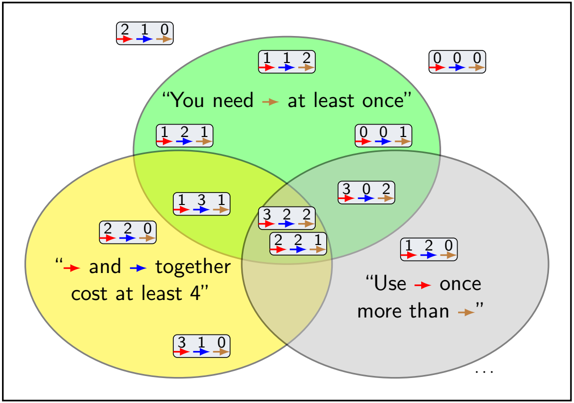


[3] Adapted from Helmert and Röger's planning course

Operator Counting heuristics

OC variables correspond to unordered actions in potential plans.[3]

Operator occurrences in potential plans

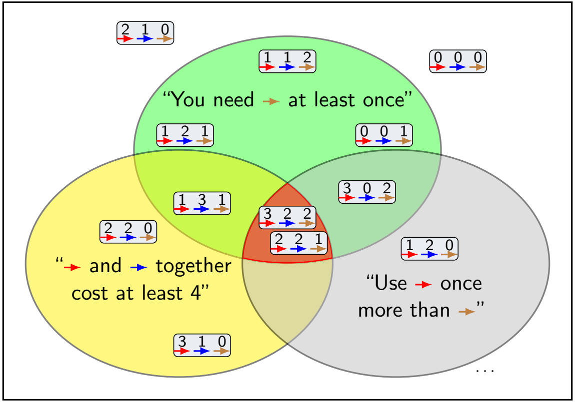


[3] Adapted from Helmert and Röger's planning course

Operator Counting heuristics

OC variables correspond to unordered actions in potential plans.[3]

Operator occurrences in potential plans

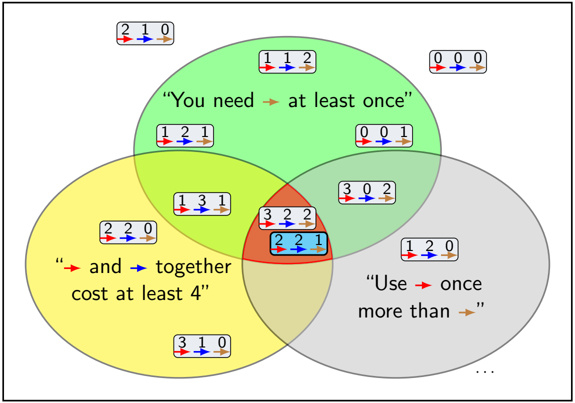


[3] Adapted from Helmert and Röger's planning course

Operator Counting heuristics

OC variables correspond to unordered actions in potential plans.[3]

Operator occurrences in potential plans



[3] Adapted from Helmert and Röger's planning course

Operator-counting Constraint

Definition: Operator-counting constraints

Let Π be a planning task with operators O and let s be a state. Let $\mathcal{V}$ be the set of variables Count_o for each $o \in O$. A linear inequality over $\mathcal{V}$ is called an **operator-counting constraint** for s of for every plan π for s setting each Count_o to the number of occurrences of o in π is a feasible variable assignment.

Operator-counting Constraint

Definition: Operator-counting constraints

Let Π be a planning task with operators O and let s be a state. Let $\mathcal{V}$ be the set of variables Count_o for each $o \in O$. A linear inequality over $\mathcal{V}$ is called an **operator-counting constraint** for s of for every plan π for s setting each Count_o to the number of occurrences of o in π is a feasible variable assignment.

So, what are typical operator-counting constraints?

- Landmarks: L landmark operator has a constraint $\sum_{o \in L} \text{Count}_o \geq 1$
- Flow heuristic: one flow constraint per atom a:
  $$[a \in s] + \sum_{o \in O : a \in \text{eff}(o)} \text{Count}_o = [a \in \gamma] + \sum_{o \in O : a \in \text{pre}(o)} \text{Count}_o$$
- Post-hoc Optimization:

Operator Counting Constraints for Goal Recognition

Motivation:

- Operator counting constraints represent knowledge about solutions
- allows **reasoning about solutions** to derive heuristic estimates

Operator Counting Constraints for Goal Recognition

Motivation:

- Operator counting constraints represent knowledge about solutions
- allows **reasoning about solutions** that **comply with additional constraints**:

Operator Counting Constraints for Goal Recognition

Motivation:

- Operator counting constraints represent knowledge about solutions
- allows **reasoning about solutions** that **comply with additional constraints**:
    - **actual** observations

Operator Counting Constraints for Goal Recognition

Motivation:

- Operator counting constraints represent knowledge about solutions
- allows **reasoning about solutions** that **comply with additional constraints**:
  - **actual** observations
  - **missing** observations

Operator Counting Constraints for Goal Recognition

Motivation:

- Operator counting constraints represent knowledge about solutions
- allows **reasoning about solutions** that **comply with additional constraints**:
  - **actual** observations
  - **missing** observations
  - **noisy** observations

Operator Counting Constraints for Goal Recognition

Motivation:

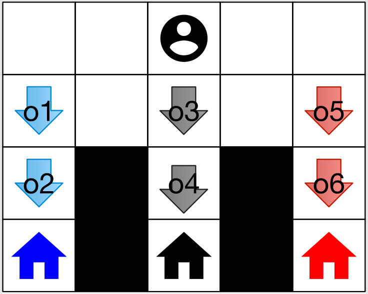
- Operator counting constraints represent knowledge about solutions
- allows **reasoning about solutions** that **comply with additional constraints**:
  - **actual** observations
  - **missing** observations
  - **noisy** observations
  - goal hypotheses
  - other constraints

Computing intersection of OC and Observations

Initial idea: compute operator-counts and compare with observations

Example

Consider an observation containing o_2 and o_3 (from a plan for g_1):



Compute $h^G(\mathcal{I})$ for:

- g_1
- g_2
- g_3

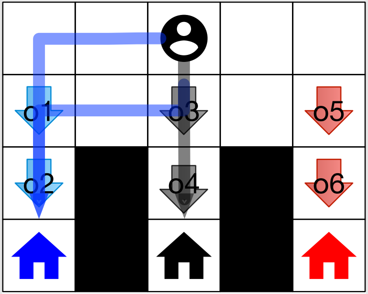
Return goal with maximum overlap

Computing intersection of OC and Observations

Initial idea: compute operator-counts and compare with observations

Example

Consider an observation containing o_2 and o_3 (from a plan for g_1):



Compute $h^G(\mathcal{I})$ for:

- g_1
- g_2
- g_3

Return goal with maximum overlap

Problem: LP has multiple (optimal) solutions

Hard Constraints for the Observations

Second idea: force OCs that comply with observations

Observation constraints

Let k_a be the number of occurrences of observations of the operator a in the sequence of Observations O for a goal recognition problem P, the hard constraint for a is:

$$\mathsf{Count}_a \geq k_a$$

We call the objective value of the resulting LP h_{HC}

Hard Constraints for the Observations

Second idea: force OCs that comply with observations

Observation constraints

Let k_a be the number of occurrences of observations of the operator a in the sequence of Observations O for a goal recognition problem P, the hard constraint for a is:

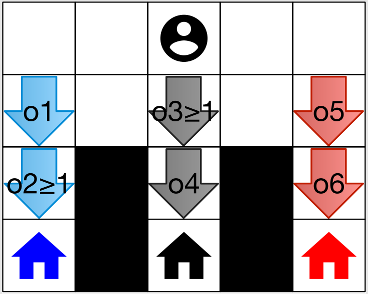
$$\text{Count}_a \geq k_a$$

We call the objective value of the resulting LP h_{HC}

h_{HC} Heuristic cost of reaching a goal G subject to the observations O

Hard Constraints for the Observations

Example: Observations o_2 and o_3 (towards g_1)

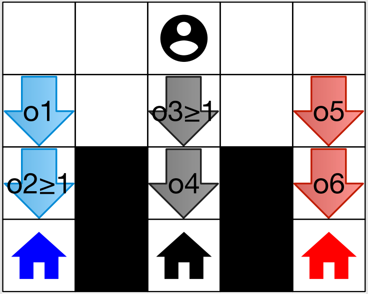


Compute $h^G(\mathcal{I})$ for:

- $h^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 3$
- $h^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 4$
- $h^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 4$

Hard Constraints for the Observations

Example: Observations o_2 and o_3 (towards g_1)



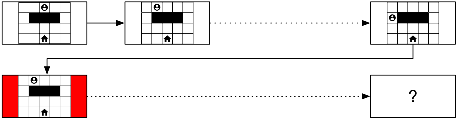
Compute $h^G(\mathcal{I})$ for:

- $h^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 3$
- $h^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 4$
- $h^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 4$

Solution: $\{G | G \in \mathcal{G} \wedge h^G_{\mathrm{HC}} \leq \min_G h^G_{\mathrm{HC}}\} = \{g_1\}$

Accounting for Uncertainty

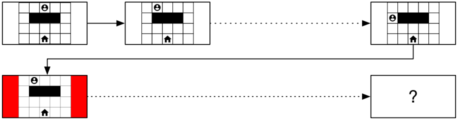
Key Challenge: Observations are unreliable in Goal Recognition



- Fast approaches (e.g. Pereira, Oren, and Meneguzzi) have a threshold to handle ties due to missing observations
- OC heuristics: **lower bound** on number of **observations**

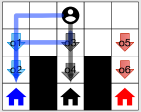
Accounting for Uncertainty

Key Challenge: Observations are unreliable in Goal Recognition



- Fast approaches (e.g. Pereira, Oren, and Meneguzzi) have a threshold to handle ties due to missing observations
- OC heuristics: **lower bound** on number of **observations**

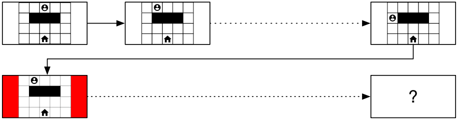
Lower bound on observations



- $h_{\mathrm{HC}}^{g_1}(\mathcal{I}) = 3$
- $h_{\mathrm{HC}}^{g_2}(\mathcal{I}) = 5$ $h_{\mathrm{HC}}^{g_3}(\mathcal{I}) = 7$

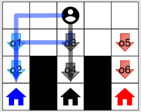
Accounting for Uncertainty

Key Challenge: Observations are unreliable in Goal Recognition



- Fast approaches (e.g. Pereira, Oren, and Meneguzzi) have a threshold to handle ties due to missing observations
- OC heuristics: **lower bound** on number of **observations**

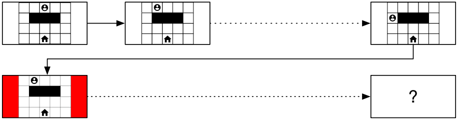
Lower bound on observations



- $h^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 3$
- $h^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 5$ $h^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 7$
- $|O| \geq 3$

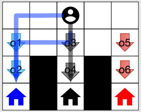
Accounting for Uncertainty

Key Challenge: Observations are unreliable in Goal Recognition



- Fast approaches (e.g. Pereira, Oren, and Meneguzzi) have a threshold to handle ties due to missing observations
- OC heuristics: **lower bound** on number of **observations**

Lower bound on observations



- $h_{\mathrm{HC}}^{g_1}(\mathcal{I}) = 3$
- $h_{\mathrm{HC}}^{g_2}(\mathcal{I}) = 5$ $h_{\mathrm{HC}}^{g_3}(\mathcal{I}) = 7$
- $|O| \geq 3$
- $U \leftarrow 1 + \frac{\min_G h_{\mathrm{HC}}^{G} - |O|}{\min_G h_{\mathrm{HC}}^{G}} = 1.33$

Solution: $\{G | G \in \mathcal{G} \wedge h_{\mathrm{HC}}^{G} \leq \min_G h_{\mathrm{HC}}^{G} * U\} = \{g_1\}$

Properties of the heuristics

Proposition 1: h_{HC} dominates h

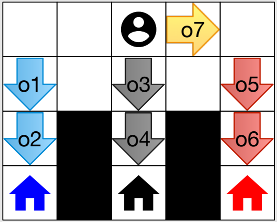
Let h be the basic operator-counting heuristic, h_{HC} be the over-constrained heuristic h_{HC} that accounts for all observations $o \in O$, and s a state of Π. Then $h_{\mathrm{HC}}(s) \geq h(s)$.

Proposition 2

The set of goals returned by h_{HC} with 100% of the observations always contains the actual goal.

Problem: Noisy observations throw off over-constrained heuristic

Observations o_2, o_3, o_7 (noisy) towards g_1

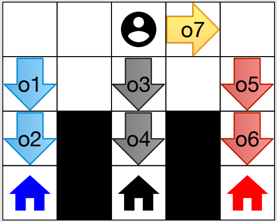


- $h^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 7$ $h^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 7$ $h^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 10$

Enforcement Delta

Problem: Noisy observations throw off over-constrained heuristic

Observations o_2, o_3, o_7 (noisy) towards g_1

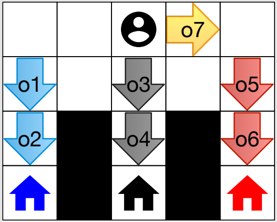


- $h_{\mathrm{HC}}^{g_1}(\mathcal{I}) = 7$ $h_{\mathrm{HC}}^{g_2}(\mathcal{I}) = 7$ $h_{\mathrm{HC}}^{g_3}(\mathcal{I}) = 10$
- $h^{g_1}(\mathcal{I}) = h^{g_2}(\mathcal{I}) = h^{g_3}(\mathcal{I}) = 2$

Enforcement Delta

Problem: Noisy observations throw off over-constrained heuristic

Observations o_2, o_3, o_7 (noisy) towards g_1

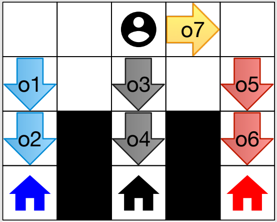


- $h^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 7$ $h^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 7$ $h^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 10$
- $h^{g_1}(\mathcal{I}) = h^{g_2}(\mathcal{I}) = h^{g_3}(\mathcal{I}) = 2$
- $\delta^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 5$ $\delta^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 5$ $\delta^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 8$

Enforcement Delta

Problem: Noisy observations throw off over-constrained heuristic

Observations o_2, o_3, o_7 (noisy) towards g_1

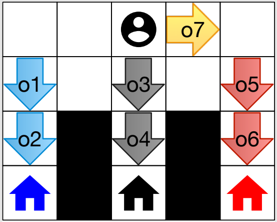


- $h^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 7$ $h^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 7$ $h^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 10$
- $h^{g_1}(\mathcal{I}) = h^{g_2}(\mathcal{I}) = h^{g_3}(\mathcal{I}) = 2$
- $\delta^{g_1}_{\mathrm{HC}}(\mathcal{I}) = 5$ $\delta^{g_2}_{\mathrm{HC}}(\mathcal{I}) = 5$ $\delta^{g_3}_{\mathrm{HC}}(\mathcal{I}) = 8$
- $U = 1.57$

Enforcement Delta

Problem: Noisy observations throw off over-constrained heuristic

Observations o_2, o_3, o_7 (noisy) towards g_1

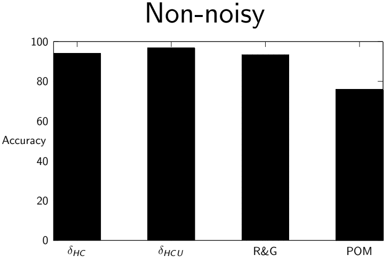


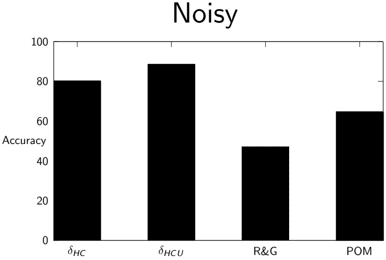
- $h_{\mathrm{HC}}^{g_1}(\mathcal{I}) = 7$ $h_{\mathrm{HC}}^{g_2}(\mathcal{I}) = 7$ $h_{\mathrm{HC}}^{g_3}(\mathcal{I}) = 10$
- $h^{g_1}(\mathcal{I}) = h^{g_2}(\mathcal{I}) = h^{g_3}(\mathcal{I}) = 2$
- $\delta_{\mathrm{HC}}^{g_1}(\mathcal{I}) = 5$ $\delta_{\mathrm{HC}}^{g_2}(\mathcal{I}) = 5$ $\delta_{\mathrm{HC}}^{g_3}(\mathcal{I}) = 8$
- $U = 1.57$

Solution: Compute OCs twice, and return min $\delta_{\mathrm{HC}}^{G} = h_{\mathrm{HC}}^{G} - h^{G}$

$$\{G | G \in \mathcal{G} \wedge \delta_{\mathrm{HC}}^{G} \leq \min_G \delta_{\mathrm{HC}}^{G} * U\} = \{g_1, g_2, g_3\}$$

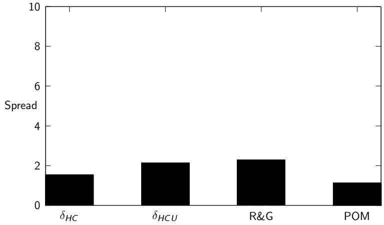
Empirical Experiments (Accuracy)





Empirical Experiments (Spread)

Non-noisy



Noisy

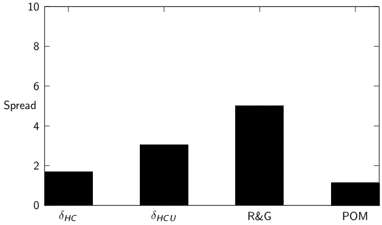


Table of Contents

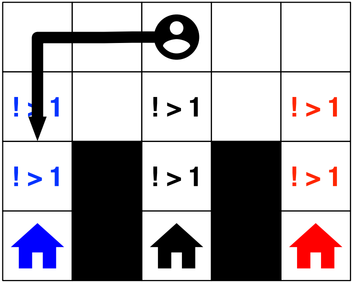
Goal Recognition using Operator-Counting Constraints

Meneguzzi, Pereira and Pereira (2020):

- Use **operator counting** information to recognize goals; and
- Operator counts and LP constraints cope explicitly with noisy observations.

Key advantages:

- **More accurate** than all previous approaches; and
- **Extensible framework** for further goal recognition work.



Future Directions

- Introduce flexibility for noise in the constraints
- Reason about all goals in one LP
- Implement approaches Efficiently

Table of Contents

Thanks and Acknowledgement

People involved in this research

- Ramon Fraga Pereira (PhD Student)
- André Grahl Pereira (UFRGS)

My hosts in Melbourne

- Nir Lipovetzky (Melbourne University)
- Miquel Ramirez (Melbourne University)

The money guys

Institutions

- Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)
  Internalization Project (PrInt)
- Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) – PQ Fellowship

Internationalization Plug

If this talk was interesting and you want to know more, talk to me:

PUCRS PrInt

`http://www.pucrs.br/print/`

Areas of work (with me) and advantages:

- Automated Planning and Goal Recognition
- Machine Learning (within reason)
- Excellent Food

Thank you!
Questions?

ESCOLA POLITÉCNICA