Plan Activity and Intent Recognition Tutorial

Sarah Keren   Reuth Mirsky   Christopher Geib

AAAI 2019

January 2019

Contents

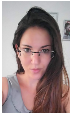
The tutorial consists of 3 main parts:

- Part 1: An historical overview of the recognition work - **Chris**
- Part 2: A recipe for specifying a recognition problem - **Sarah**
- Part 3: State of the art in recognition literature - **Reuth**

A little bit about us before we start

 **Chris** - Ph.D UPenn, UBC, Honeywell, UEdin, Drexel, SIFT.

 **Sarah** - Ph.D Technion, now Postdoc at Harvard.

 **Reuth** - Ph.D. Ben-Gurion, soon Postdoc at UT Austin.

We will most probably have different answers to any of your questions !

Contents

Part 1:

An historical overview of the recognition work - Chris

We were there (almost) at the beginning.

- 1956 - Dartmouth conference occurs (coins the Term AI)
- 1959 - Newell, Shaw, Simon build the General problem-solver
- 1971 - STRIPS, Fikes and Nilsson.
- 1978 - The Plan Recognition Problem: An intersection of psychology and artificial intelligence. C. Schmidt, N. Sridharan, J. Goodson, *Artificial Intelligence*, vol 11, pp. 45-83, 1978
- ... and don't get me started on MDP/HMMs ('30s and '40s)

Schmidt et. al's Def.

- "The problem of plan recognition is to take as input a sequence of actions performed by an actor and to infer the goal pursued by the actor and also to organize the action sequence in terms of a plan structure. This plan structure explicitly describes the goal-subgoal relations among its component actions."

But it was harder than we thought.

- Like many of the other (now) sub-fields of AI, we realized
  1) Formalizing the problem as a logic wasn't gonna cut it, and
  2) there was more than one problem.
- At least three different problems that were at one time or another called plan recognition.
  - Activity Recognition
  - Goal Recognition
  - Plan Recognition

Activity Recognition

Definition:

- INPUT: A sequence of noisy sensor inputs over time.
- OUTPUT: A unique label for each temporal subsequence.

Central problem:

Dealing with noise in the input observation stream.

Alternative characterization:

Classification/Labeling of noisy temporal observations.

Example:

- Video segmentation of football plays. (e.g. passing, clearing, throw-in, etc...)

Goal Recognition

Definition:

- INPUT: An ordered sequence of discrete symbolic input tokens.
- OUTPUT: A unique label (perhaps with a probability) for each temporal subsequence.

Central problem:
Dealing with evidence for multiple conflicting hypothesis.
Alternative characterization:
Classification/Labeling temporal observations where each observation can contribute to many possible labels.
Example:

- Identifying computer user goals from observing their actions. (e.g. searching web, starting a new document, confused, etc...)

Plan Recognition

Definition:

- INPUT: An ordered sequence of discrete symbolic input tokens.
- OUTPUT: Complex structure capturing plan being executed. Potentially including abstract tasks that have been done and which are yet to do and traditionally the goal of the plan.

Central problem:

Combining sequences of lower level observations into larger structured patterns. (probabilistic or not)

Alternative characterization:

Temporal pattern matching, sequence matching.

Example:

- Identify the plan and goal of cyber intruders and their progress though a network. (e.g. Bragging, DoS, espionage, what machines do they pwn?...)

Chris' Digression on "Intent Recognition"

Don't use this term. Nothing good comes from it.

- If you hear/read it, ask yourself what they are actually doing.
- People that say this DON'T mean a plan, action, or state.
- Some things they might mean:
  - **Civilians (non-AI researchers)**: Ineffable magic that differentiates human's and synthetic agent's actions.
  - **Some philosophers**: A separate pro-attitude towards a plan denoting a commitment to its execution.
  - **Other philosophers**: A mental state in which an agent believes a sequence of actions will cause a state they desire and believes that they will execute those actions to that end.
  - **Military**: What the Sr. Officer wanted to have happen.

More domains than you can imagine.

Natural language understanding, discourse processing, video segmentation, video games, assistive care for the elderly, process control systems for manufacturing, software help systems, computer network security, insider threat detection, international planning competition domains, cooking, cognitive orthotics, capture the flag, etc...

Disclaimers Before We Go Further

This is NOT everything you should know about...

- these papers!
- history of plan rec!
- history of goal rec!
- history of activity rec!

Cited papers may and or may not be the most famous piece of work.

Read the literature. You might see something else.

Kautz[1]: PLAN REC

- Domain: Cooking.
- Approach: Graph covering based on a preexisting plan library.
- Core contribution: Plan libraries and formalization.
- Limitation: Assumed that a minimal graph covering was the best (ie.non-probabilistic)

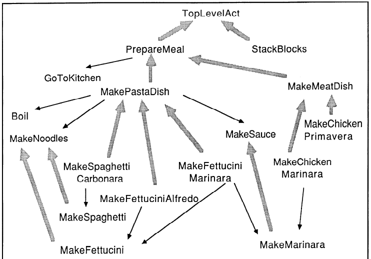


Figure 1: Action Hierarchy

[1] H. Kautz and J. Allen, Generalized plan recognition, *Proceedings of the National Conference on Artificial Intelligence*, pp. 32-38, 1986.

Carberry[2] and Littman [3] : PLAN REC? GOAL REC?

- Domain: Natural Language Understanding, Discourse.
- Approach: Logical inference using the situation calculus.
- Core contribution: Formalizing the vast inference in language. (Grosz, Pollack, Allen...)
- Limitation: Cost of encoding and inference

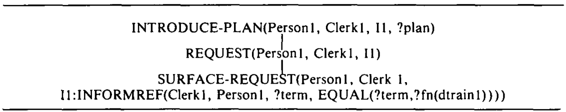


Figure 8. Chaining produces an intermediate plan recognition structure.

[2] S. Carberry. *Plan Recognition in Natural Language Dialogue.* ACL-MIT Press Series in Natural Language Processing. MIT Press, 1990.

[3] D. Littman and J. Allen, A plan recognition model for subdialogues in conversation, *Cognitive Science*, vol. 11(2), pp. 163-200, 1987.

Vilain[4]: PLAN REC

- Domain: Complexity analysis
- Approach: Parsing formal plan grammars.
- Core contribution:
  - 1) Complexity results,
  - 2) Need for generativity.
- Limitation: No actual system provided.
- Key results:
  - "Recognizing plans with abstraction and partial step order is NP-complete,..."
  - "An acyclic hierarchy does not contain any recursive plan definitions, and could in fact be encoded as a regular (finite-state) grammar."

[4]M. Vilain, Getting serious about parsing plans: A grammatical analysis of plan recognition, *Proceedings of AAAI-90*, pp. 190-197, 1990.

Goldman[5]: PLAN REC

- Domain: Story understanding.
- Approach: Dynamically assembly of Bayes nets.
- Core contribution: The problem of universal quantification and unbound vars in stories.
- Limitation: Limited by Bayesian methods of the time, and building Bayes nets dynamically.

[5] R. Goldman and E. Charniak. Probabilistic text understanding. In *Statistics and Computing*, Vol 2, pp. 105-114, 1992.

Pynadath[6]: PLAN REC

- Domain: Driving/Lane Selection.
- Approach: Probabilistic State Dependant Grammar to specialized probabilistic inference
- Core contribution: Formalizing the problem in a grammar looked at a dynamic Bayes net.
- Limitation: Then built their own probabilistic algorithm.

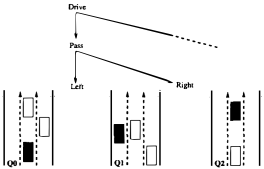


Figure 2: Simple PSDG parse tree from traffic domain.

[6] D. Pynadath and M. Wellman, Accounting for context in plan recognition with application to traffic monitoring, *Proceedings of UAI-95*, pp. 472-481, 1995.

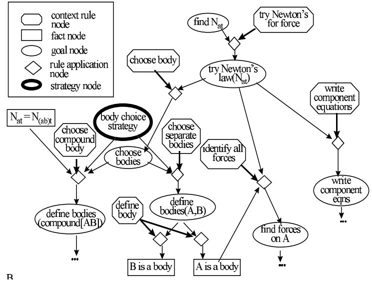
Horvits[7]: GOAL REC

- Domain: Software assistive systems
- Approach: Bayes nets.
- Core contribution: Goal recognition in a real system.
- Limitation: Limited by Bayesian methods of the time. And Bayes nets of our time. :-)

[7]E. Horvitz, J. Breese, D. Heckerman, D. Hovel, and K. Rommelse, The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users, *Proceedings of UAI-98*, pp. 256-265, 1998.

Conati[8]: GOAL REC

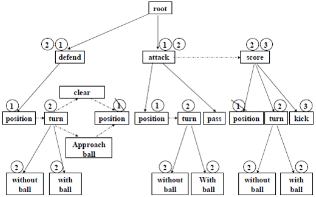
- Domain: Educational agents.
- Approach: Bayes nets.
- Core contribution: Explicit modeling of incorrect plans.
- Limitation: Bayesian models: propositional and scale.



[8] C. Conati, A. Gertner, K. VanLehn, and M. Druzdzel, On-Line student modeling for coached problem solving using Bayesian networks, *Proceedings of User Modeling-97*, pp. 231-242, 1997.

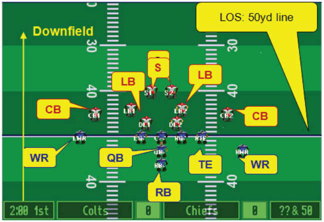
Avrahami-Zilberbrand[9]: PLAN REC

- Domain: RoboCup
- Approach: Marker passing over packed parse trees.
- Core contribution: More efficient plan recognition as parsing.
- Limitation: Multiple instances of the same plan.



[9] D. Avrahami-Zilberbrand and G. Kaminka, Fast and complete symbolic plan recognition, *Proceedings of IJCAI-05*, pp. 653-658, 2005.

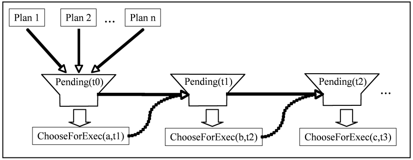
- Domain: Video games (RUSH 2008 football).
- Approach: Support vector machines for classification
- Core contribution: Real-world deployment fast enough to make a difference.
- Limitation: New plays?



[10] K. Laviers, G. Sukthankar, D. Aha, M. Molineaux, and C. Darken, Improving Offensive Performance Through Opponent Modeling, *Proceedings of AIIDE*, pp.58-63, 2009.

Geib[11]: PLAN REC

- Domain: Synthetic domains.
- Approach: Parsing of probabilistic plan recognition as parsing.
- Core contribution: Efficient grammars for parsing, multiple concurrent goals, pending sets.
- Limitation: Required building the complete set of parses.



[11] C. Geib and R. Goldman. A probabilistic plan recognition algorithm based on plan tree grammars, *Artificial Intelligence*, vol 173(11), pp. 1101-1132, 2009.

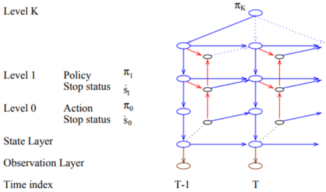
Ramirez[12]: GOAL REC

- Domain: IPC domains.
- Approach: Plan recognition as planning.
- Core contribution: Use of planning algorithms.
- Limitation: Initial work was actually doing goal recognition.

[12] M. Ramírez, and H. Geffner, Plan recognition as planning. in *Proceedings of IJCAI*, pp. 1778-1783, 2009

Bui[13]: PLAN REC

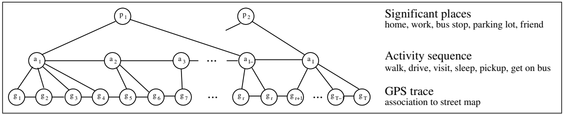
- Domain: 2D navigation.
- Approach: Hierarchical Hidden Markov Models.
- Core contribution: Using HMMs at multiple levels to actually do plan recognition
- Limitation: Fully ground models.



[13] H. Bui, S. Venkatesh, and G. West. Policy recognition in the Abstract Hidden Markov Model, *Journal of Artificial Intelligence Research*, vol 17, pp. 451-499, 2002.

Liao [14]: ACTIVITY REC

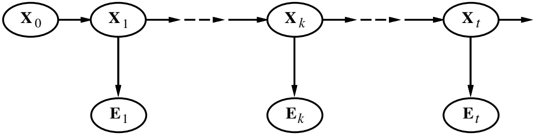
- Domain: Daily activity tracking. (2D tracking)
- Approach: Hierarchical conditional random fields.
- Core contribution: Using HCRF, and real GPS data
- Limitation: location based...



[14] L. Liao, D. Fox, and H. Kautz. Hierarchical conditional random fields for GPS-based activity recognition, *Proceedings of the 12th International Symposium of Robotics Research (ISRR)*, 2005.

A Small Nod to the Markov Decision Process People

- In a sense they were here first.
- If you have a Hidden Markov Model:
  - *Filtering*: $P(X_t|e_{1:t})$ predicting the current hidden state.
  - *Prediction*: $P(X_{t+k}|e_{1:t}k > 0$ predicting the next hidden state.
- What if the hidden state captured the possible plan states?



Contents

Part 2:

A recipe for specifying a recognition problem - **Sarah**

Formulating a recognition problem

In this part of the tutorial we will focus on the following question:

What are the elements that need to be specified when defining a recognition problem ?

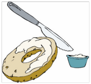
As a running example we will use the human-robot collaboration setting by Levine and Williams (ICAPS'14 and JAIR'18)

Running example: Human Robot Collaboration

A person named Alice is making breakfast for herself with the help of her trusty robot. Alice intends to either make coffee (for which she uses a mug) or get some juice (for which she uses a glass). In addition, Alice is having either a bagel with cream cheese or some cereal and milk.

The robot is trying to detect Alice's intentions, so it can assist her by, for example, getting the utensils she needs to complete a specific task.



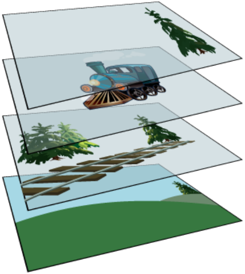



In the original formulation, Alice is running late for work, so she her food and food and drink must be ready within 7 minutes. However- we are focusing today on recognition.

Elements of a Recognition Problem

- Environment
- Acting agent (actor)
- Recognition system (recognizer)

I (Sarah) like to think of them as stacked layers



Environment

The setting in which agents act (a.k.a as the *domain theory*)
Can be described as a tuple $E = \langle S, I, A, T, \mathcal{G} \rangle$ with

- State space S
  - often, a set of features F is used to describe a state
- Set of possible initial states I
- Set of actions A(s) that can be performed at each state
  - deterministic / stochastic actions
  - temporal actions
- Transition function T
  - deterministic: $T : S \times A \to S$
  - non-deterministic: $T : S \times A \to 2^{|S|}$
  - stochastic: $T : S \times A \times S \to [0, 1]$
- Set of possible goals $\mathcal{G}$ (states or conditions to be met)

We may also have constraints (e.g. temporal constraints) that need to be respected

Environment E

Plan π

A complete sequence $a_0, a_1, \ldots, a_n$ of actions that takes an agent from a (initial) state to a (goal) state.

History h

A sequence of state transitions $s_0, s_1, \ldots, s_n$ from a (initial) state to a (goal) state.

Execution e

A sequence of state-action transitions $s_0, a_0, s_1, a_1 \ldots, s_n, a_n, s_{n+1}$ from a (initial) state to a goal.

Prefix: in recognition, we may consider the prefixes of each of the above elements

Environment

- In continuous domains, we may have numeric-valued features. Actions transition from one state to another via paths through the state space, rather than through discrete states (Vered & Kaminka, 2017)
- Policy: mapping from states to actions (plan as a special case)
    - Typically used to represent partially observable or stochastic environments

Policy/ Plan Set

An environment induces a set Π of paths / policies that represent the set of possible behaviors in the environment.

Environment: Running Example

In our running example:

- A state specifies, the position of the objects (e.g. mug_on_table), the status of the different sub-tasks, etc.
- Temporal actions represent the sub-tasks of breakfast preparation, e.g., pour coffee.
- Transitions can be represented as deterministic or probabilistic (e.g., an action may fail with some probability)
- The agent's objective is to prepare a breakfast comprised of a drink (juice or coffee) and food (bagel or cereal).
- Possible plans
  - Get mug, pour coffee, get bagel, toast bagel ...
  - Get bagel, toast bagel, get glass ...
  - ...




Actor

The *actor* (acting agent) specifies the **assumptions made** w.r.t. how an agent with a specific goal chooses to behave in a given environment.

- In all our settings, we are assuming agents enter the environment and follow a policy / plan to achieve some goal.
- Agent behavior is influenced by the actor's:
  - familiarity with the environment (possibly reflected by its sensor model)
  - capabilities and preferences (e.g., can they compute an optimal plan?)
  - relationship to the recognizer
- Note: recognition in a multi-agent setting is an interesting extension but beyond scope for today!

Actor

When we model the actor, we need to account for the set of plans an actor may follow to achieve each of the possible goals

In particular, we need to answer the following questions:

- **How does the actor make decisions?**
- **What does the actor know and how does it perceive its surrounding ?**
- **What is the actor's relationship to the recognizer?**
- **What is the best way to represent the actor?**

Remember!

We are representing the actor from the recognizer's point of view

I (Sarah) like to think about this is the actor's Decision Making Mechanism: mapping $\theta : \mathbb{B} \times \mathcal{G} \rightarrow 2^A$ from (belief) states to actions

Actor

Possible answers to our questions about the actor:

- **How does the actor make decisions?**
  - For example: actors are optimal or sub-optimal
- **What does the actor know and how does it perceive its surrounding ?**
  - For example: when partially informed, we need to account for the actor's sensor model.
  - Typically, a belief state is used to represent the states an agent deems as possible / a probability distribution over states.

Actor

Possible answers to our questions about the actor:

- **What is the actor's relationship to the recognizer?**
  - Agnostic - the actor is agnostic to / unaware of the recognition process
  - Adversarial - the actor wants to deceive the recognizer (given its own constraints)
  - Intended - the actor wants to implicitly communicate its goal / plan to the recognizer

Strongly related to the topic of **explainable/ privacy preserving planning** - which assumes the role of an agent that chooses to behave in a way that **reveals / obfuscates** its objective

Representing the actor: Common Formulations

Possible answers to our questions about the actor:

- **What is the best way to represent the actor?**
  Two commonly used representations: plan libraries and domain theories

  Plan libraries:

  - Hierarchical Task Network (HTN)
  - Formal Grammars
  - AndOr Trees

  Domain theory (planning):

  - Plan recognition as planning

Common Formulations: plan libraries

Hierarchical Task Network (HTN)

- HTN representation of the environment contains methods, where each method
  - includes a prescription for how to decompose some task into a set of subtasks
  - restrictions for the task's applicability.
  - constraints on the subtasks and the relationships among them
- Planning by task decomposition - HTN planning works by expanding tasks and resolving conflicts iteratively, until a conflict-free plan can be found that consists only of primitive tasks.

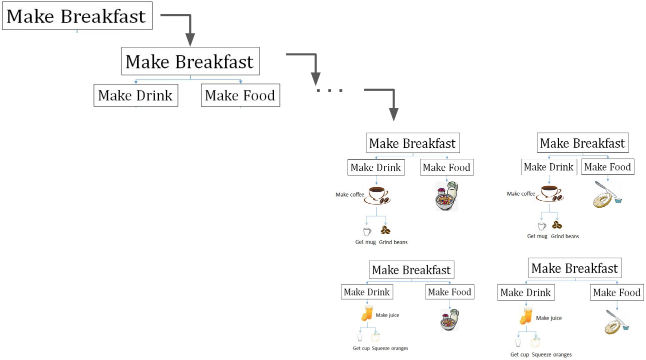
Common Formulations: HTN

Hierarchical Task Network (HTN)[15]

- The language $L = \langle V, C, P, F, T, N \rangle$ where
  - V,C,P are sets of variable, constant and predicate symbols (respectively)
  - F - primitive-task symbols (denoting actions)
  - T - compound-task symbols
  - N - labels
- L used to construct a task network of the form $[(n_1 : \alpha_1)...(n_m : \alpha_m), \phi]$, where
  - $\alpha_i \in F \cup T$.
  - n_i is a label for α_i (to support multiple occurrences of α_i)
  - ϕ is a Boolean formula representing constraints (temporal, ordering, etc.)
- A method is a pair $\langle NA, TN \rangle$ where $NA \in T$ and TN is a task network.
- Methods specify how to accomplish the subtasks of a non-primitive task
  - In some formulations a method can have preconditions on its application
- A *planning domain* D is a pair $\langle F, Me \rangle$, where F is a list of operators (one for each primitive task), and Me is a set of methods
- A *planning problem* P is a tuple $\langle D, I, P \rangle$ with planning domain D, initial state I, and task network D
- A solution is a plan, a sequence of ground primitive tasks

[15] Erol, Hendler & Nau 1994

Common Formulations: HTN



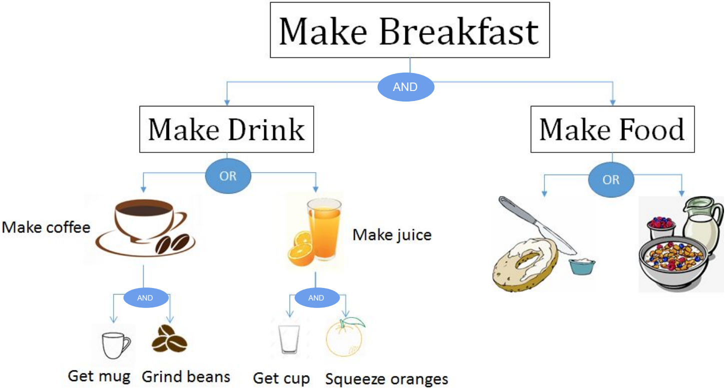
Common Formulations: AND-OR

AND-OR trees [16]

- AND-nodes represent methods for achieving a particular task
  - all of the children of an AND-node must be performed in order to perform the parent task
  - a partial order may be imposed by annotating them with pairwise ordering constraints
- OR-nodes represent choice nodes where the agent may choose one of a number of alternate methods to achieve a task
  - only one of the children of an OR-node needs to be performed in order for the parent action to be achieved
  - for this reason, ordering constraints between the children of an OR-node are not allowed.

[16] Geib & Goldman 2009

Common Formulations: AND-OR



(AND (OR (AND get-mug grind-beans)(AND get-cup squeeze-juice)) (OR make-bagel make-cereal)) . . .

Common Formulations: Grammars [17]

- Based on grammatical formalisms developed for use in natural language parsing(NLP)
- Plan lexicon - defines the plans to be recognized
- Production rules represent the relationship between a compound task and its sub-tasks
  - **Make Breakfast $\rightarrow$ Make Food, Make Drink**
- Lexicalized grammar: every production rule has at least one distinguished terminal in the right hand side.
- To perform recognition, observations are parsed using one or more plan structures meeting the requirements in the lexicon
- Plans are described as plan trees, with the root as the goal and the primitive actions as leaves

[17] Geib& Goldman 2009

Common Formulations: Grammar

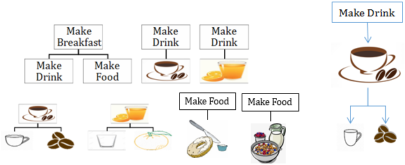
There are different representations of grammars for recognition. A Formal Grammar is defined[18] as a 4-tuple $\langle Nt, \Sigma, P, S \rangle$ where

- Nt is a finite set of nonterminal symbols
- Σ is a finite set of terminal symbols disjoint from Nt
- P is a set of production rules that have the form $A \to \alpha$ where $A \in Nt$ and $\alpha \subseteq Nt \cup \Sigma$.
  - Some grammars include ordering constraints, in which case a production rule may have the form $A \to \alpha, \omega$ where ω is s partial order of α. This changes the expensiveness of the grammar.
- S is the start symbol (represented as the set of goals $G \subseteq Nt$ in our context).

There are many other representations, each with its pros and cons

[18] Hopcroft, Motwani, and Ullman 2006

Common Formulations: Grammar



- Σ = { Get cup, Squeeze Oranges, Get Mug, Grind Beans, Bagel, Cereal }
- NT = {Make Breakfast, Make Drink, Make Food, Make Coffee, Make Juice }
- P = {
  - Make Breakfast $\rightarrow$ Make Food, Make Drink
  - Make Breakfast $\rightarrow$ Make Drink, Make Food
  - Make Drink $\rightarrow$ Make Coffee
  - Make Drink $\rightarrow$ Make Juice
  - Make Coffee $\rightarrow$ Get Mug, Grind Beans . . . }
- G = { Make Breakfast }

Common Formulations: plan and goal recognition as planning [19]

- Recognition uses a domain theory to represent the actor's behavior.
- A generative approach that can be used to rank the different goals according to a probability distribution $P(G|\vec{o})$, where $\vec{o}$ is the observed action sequence
- A key benefit is the ability to use off the shelf solvers (and optimal planners in particular) to represent the actor's behavior.
- The framework was first suggested for deterministic settings (Ramirez & Geffner 2009,2010) and later extended to support stochastic settings (Ramirez & Geffner 2011)

[19] Ramirez & Geffner 2010

Common Formulations: plan and goal recognition as planning [20]

The actor is represented by a planning problem $P = \langle F, I, A, G \rangle$ where

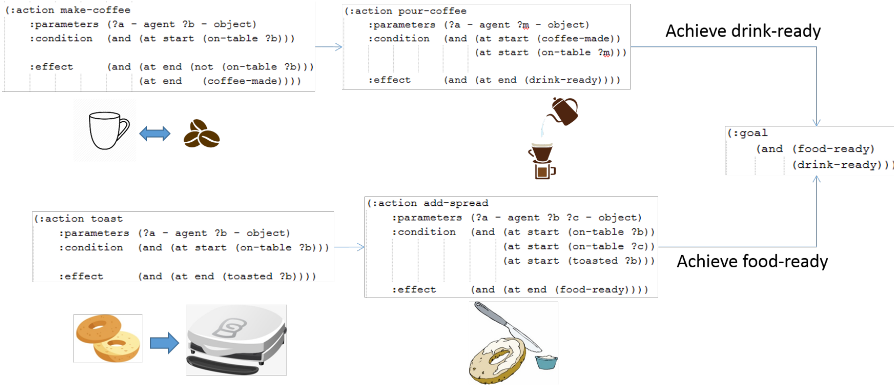
- F is the set of features
- I is the initial state
- A is the set of deterministic actions
- G is the goal

The corresponding recognition problem is a tuple $\langle P, \mathcal{G}, \vec{o}, Prob \rangle$ where

- P is the planning problem
- $\mathcal{G}$ is the set of goals
- $\vec{o}$ is the observation sequence (of actions) observed
- $Prob$ is the prior probability distribution over the goals.

[20] Ramirez & Geffner 2010

Common Formulations: plan and goal recognition as planning



Actor: notes on the formalisms

- There are many representations we did not cover.
- Regardless of which representation is used, in the context of recognition, it is used to model the assumption the recognizer makes about the actor, who may be using a different model.
- Ramirez and Geffner (2016) showed that the domain formulation is equivalent to the plan library approach for libraries with finite yield.
  - Libraries can be compiled into strips theories, however
  - Compilation requires bounding the depth of the plan derivation.
  - Produces a library with only finite yield.

Recognizer (Recognition System)

The actor's model specified how the recognizer expects the actor to behave w.r.t each goal/ plan / activity.

For the recognizer, we need to specify the following.

- **Observability** - How does the recognizer perceive the actor's behavior? What is the recognizer's sensor model
- **Objective** - What is the recognizer's objective ?
- **Possible Interventions** - Can the recognizer interact with the actor or affect its behavior?

Recognizer's Observability

- The recognizer's sensor model is a mapping from executions/ plans / sequences to observation sequences (e.g., $O_{rec} : E \rightarrow \vec{o}$)
- Typically defined using a mapping from actions / states to observation tokens $O_{rec} : A \rightarrow O$
- The observation sequence $\vec{o}$ is the entity that is analyzed

Recognizer's Objective

In all our settings, we are assuming agents enter the environment and follow a policy / plan to achieve some goal.

Three types of Recognition

- **Plan recognition** - identify the sequence of actions the actor follows to achieve it's goal
- **Goal recognition** - identify the end conditions the actor wishes to a achieve
- **Activity recognition** - identify a specific action that is being performed by the actor

Recognizer's Objective

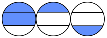
The recognition task

- Typically, the recognizer wants to recognize the actor's goal / plan / activity as soon as possible.
- The recognition task can be generally characterized via:
  - $P(\pi|\vec{o})$ for plan recognition, where π is a complete plan,
  - $P(G|\vec{o})$ for goal recognition, where $G \in \mathcal{G}$ is the goal,
  - $P(a|\vec{o})$ for activity recognition, where a is an activity,

  where $\vec{o}$ is the perceived observation sequence.
- As a special case, the mappings above can be deterministic.
- The recognition result is some aggregated account of the individual measures.
- Typically, the objective is to find the most probable goal / plan / activity.

The relationship between plan, activity and intent recognition

- As we heard, in the past, each task was treated separately.
- We found it hard to find a clear distinction.
- **It's a matter of perspective:** the type of recognition is defined w.r.t. to the specific setting. Activity recognition in one setting is the goal recognition in another.



Possible Intervention

The recognizer may have a way to affect the actor's behavior.

- Offline
- Online
- Direct communication

Intervention

Offline design: **Goal Recognition Design** (Keren, Gal and Karpas - ICAPS 2014)

- Minimizing the **Worst Case Distinctiveness (wcd)** the maximal actor progress before recognition is guaranteed.
- Given a set of possible modifications (e.g., disallowing actions, sensor improvements), and a set of design constraints, what is the best way to change the environment to minimize *wcd*.
- First model accounted for optimal agents in fully observable environments, but later extended to support stochastic setting, sub-optimal agents and noisy recognizer sensor models (Keren et al., 2015; 2016a; 2016b; 2018; Wayllace et al., 2016; 2017)

Intervention

Online: Bisson, Kabanza, Benaskeur & Irandoust 2011

- Provoking the actor to behave in a specific way by setting the value of environment features.
- Events in which propositions are made true or false by another agent.
- Provoking an event may cause the opponent to react upon it, thus revealing his intended behaviour / goal.
- Deciding when and which event to provoke is treated as a planning problem.

Intervention

Direct Communication: Sequential Plan Recognition (Mirsky, Stern, Gal, Kalech 2018)

- Breaking the keyhole recognition paradigm
- Asking the actor questions about its plans / goals
- Reasoning about information gain of possible queries

Checklist

- ✓ What are the dynamics of the environment ?
- ✓ What are the assumptions made on actor behavior ?
- ✓ What is the language used to represent actor behavior ?
- ✓ What is the relationship to the recognizer ?
- ✓ How is the actor's behavior perceived ?
- ✓ What are the possible interventions ?
- ✓ What is the recognition objective ?

Contents

Part 3:

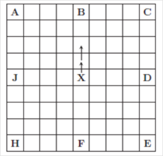
Solution Approaches: State of the art in recognition literature - Reuth

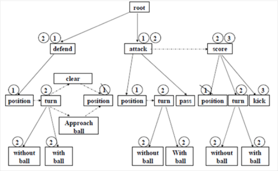
Solutions approaches

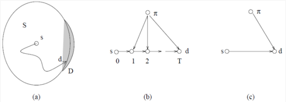
Planning $(\langle A, S, G, R \rangle)$ focuses on the fact that the actor moves from state to state and changes the environment.

Parsing $(A \rightarrow \alpha)$ focuses on the fact that a plan is constructed hierarchically in the actor's mind.

Policies $(\Pi : A \rightarrow N)$ focus on the fact that execution and observing an execution might not be deterministic.



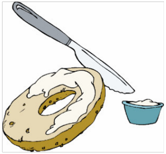




Ramírez and Geffner 2013 Avrahami-Zilberbrand and Kaminka 2009 Bui, Venkatesh and West 2002

Use case - Breakfast (adapted from Levine and Williams 2018)

Alice is making breakfast for herself with the help of her trusty robot. The team is either making coffee (for which Alice uses a mug, and for which the coffee beans need to be grounded) or getting some juice (for which Alice uses a glass, and oranges need to be pressed). To eat, the team is either making a bagel with cream cheese or getting some cereal and milk. [21]



[21] The original formulation takes into account temporal constraints.

Plan Recognition as Planning

```
(:action make-coffee
    :parameters (?a - agent ?b - object)
    :condition  (and (at start (on-table ?b)))

    :effect     (and (at end (not (on-table ?b)))
                     (at end   (coffee-made))))
```

```
(:action pour-coffee
    :parameters (?a - agent ?m - object)
    :condition  (and (at start (coffee-made))
                     (at start (on-table ?m)))
    :effect     (and (at end (drink-ready))))
```

Achieve drink-ready

```
(:goal
    (and (food-ready)
         (drink-ready))))
```

```
(:action toast
    :parameters (?a - agent ?b - object)
    :condition  (and (at start (on-table ?b)))

    :effect     (and (at end (toasted ?b))))
```

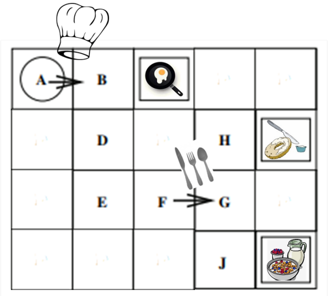
```
(:action add-spread
    :parameters (?a - agent ?b ?c - object)
    :condition  (and (at start (on-table ?b))
                     (at start (on-table ?c))
                     (at start (toasted ?b)))

    :effect     (and (at end (food-ready))))
```

Achieve food-ready

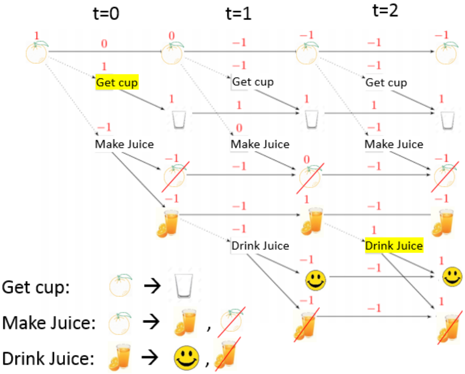
Plan Recognition as Planning (Ramirez and Geffner 2009,2010)

- Possible Goals (G): $\{at(C)\}, \{at(I)\}, \{at(K)\}$
- Observations (O): arrows
- $\forall g_i \in G$: $L(g_i \mid O) = C_i(O) - C_i(\neg O)$
  - $C_i(O)$ - the cost of reaching g_i while going through O
  - $C_i(\neg O)$ - the cost of reaching g_i without going through O
- $p(g_i \mid O) \cong \frac{1}{e^{\beta \cdot L(g_i \mid O)}+1}$



A Fast Goal Recognition Technique Based on Interaction Estimates (Martin, Moreno and Smith, 2015)

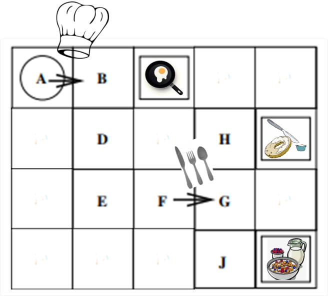
- Instead of running a planner to calculate $cost(g_i)$, it calculates the cost interaction of two or more actions
- For a goal g_i with predicates $g_i^1 \ldots, g_i^n$: $cost(g_i) \cong \sum_{j=1}^{n}[cost(g_i^j) + \sum_{k<j} I(g_i^j, g_i^k)]$



(t=0) Get cup
(t=2) Drink Juice

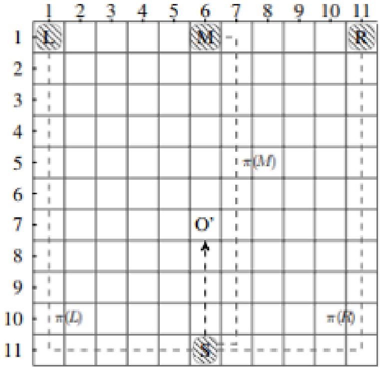
Plan Recognition as Planning Revisited (Sohrabi, Riabov and Udrea, 2016)

- Comparing to the k-best plans (or diverse plans) for each goal $g_i \in G$
- Reasons about noisy and missing observations
- Given O and g_i, the cost of plan π that meets g_i and satisfies O is: $cost_{g_i,O}(\pi) = cost(\pi) + b_1 M_{g_i,O}(\pi) + b_2 N_{g_i,O}(\pi)$ where $M_{g_i,O}(\pi)$ is number of missing obs and $N_{g_i,O}$ noisy obs



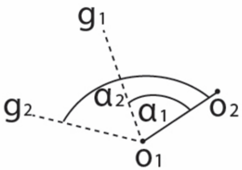
Cost-based Goal Recognition for Path-Planning (Masters and Sardina, 2017)

- Reasons about offline vs. online computation time
- Improves original formula (except for one special case): instead of $C_i(\neg O)$, uses C_i which is independent of O
- $\forall g_i \in G$:
  $L(g_i \mid O) = C_i(O) - C_i$



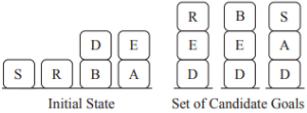
Heuristic Online Goal Recognition in Continuous Domains (Vered and Kaminka, 2017)

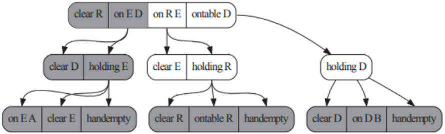
- Enhancing PRaP to continuous domains
- Proposes two heuristics inspired from mirroring neurons:
  - RECOMPUTE - recomputes new plans only if the new observations seems to change the plan significantly
  - PRUNE - prunes unlikely goals (reduces $|G|$)



Landmark-based Heuristics for Goal Recognition (Pereira, Oren and Meneguzzi, 2017)

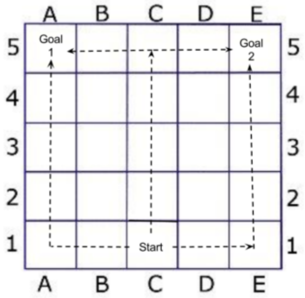
- Uses landmarks to improve runtime
- Heuristic 1: Estimate proximity to each goal (what is the ratio between achieved and not-achieved landmarks)
- Heuristic 2: Add weights to landmarks according to their uniqueness



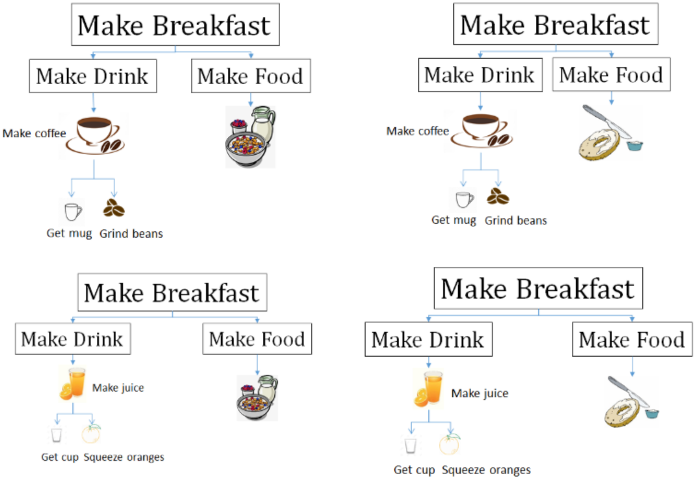


Goal Recognition Design (Keren, Karpas and Gal, 2014)

- Design for facilitating online goal recognition
- Offline analysis of the domain - defined **worst case distinctiveness** *wcd*.
- Domain design helps to reduce *wcd*.

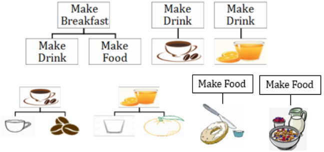


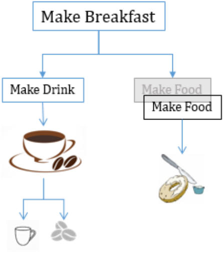
Plan Recognition as Parsing

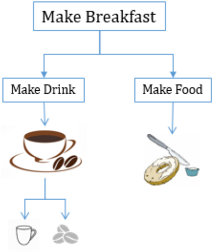


PHATT : Probabilistic Hostile Agent Task Tracker (Geib and Goldman, 2005)

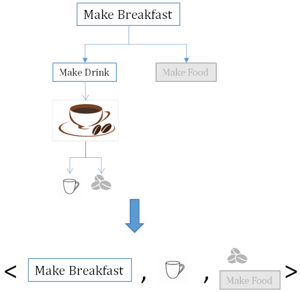
- Input 1: plan libraries as a set of recipes with partial ordering (Make Breakfast $\rightarrow$ Make Drink, Make Food $|$ ϕ)
- Input 2: observation sequence ($\langle$  ,  $\rangle$)
- Output: set of explanations
- In Figures: Set of recipes; Combining leftmost trees; Explanation

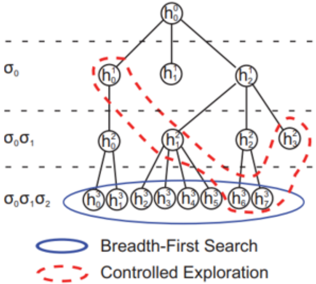






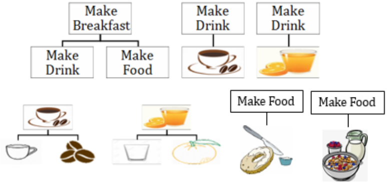
- Algorithms that use string rewriting instead of plan trees
- Output goals and frontier, do not output plan decomposition



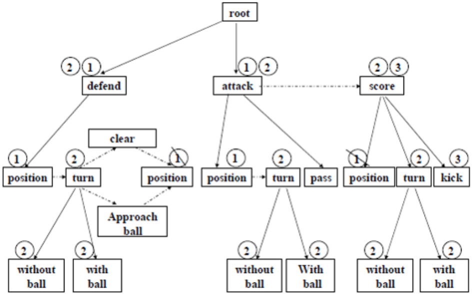


YR (Maraist, 2017)

- $LR(0)$ parser which uses string shuffling in various levels of the plan library
- At any point in the parsing, can use the shuffle operator to get a shuffling of the parsed string
- Can parse  using a DFA

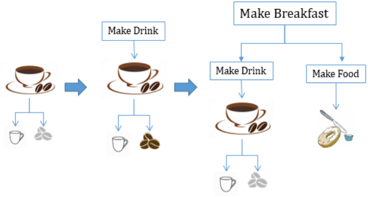


- Uses a single structure to represent the plan library
- Each observation adds marks on nodes it can be mapped to
- Upon request, can answer "where are you now?" (current state query) or "what is the path you took?" (history state query)

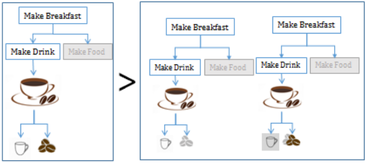


SLIM (Mirsky and Gal, 2016) & CRADLE (Mirsky and Gal, 2017)

- Both were developed to handle real-world problems
- SLIM combines top-down and bottom-up parsing



- CRADLE prunes explanations according to heuristics based on human behaviour (e.g., coherency of plans)



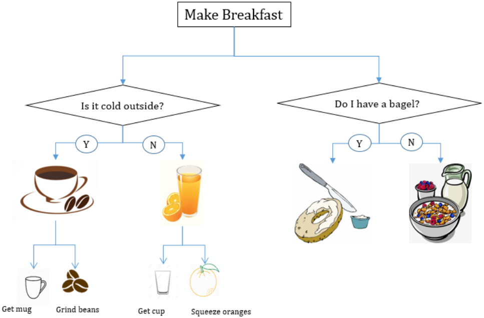
ELEXIR (Geib, 2009)

- Combines YAPPR's compactness with PHATT's expressibility
- Using Combinatory Categorial Grammars (CCGs)
- Instead of Make Breakfast $\rightarrow$ Make Drink, Make Food | (Make Drink $<$ Make Food), etc. use: Grind Beans $\rightarrow$ (Make Breakfast / Make Food) \ Mug | (Make Breakfast / Make Food) / Mug.

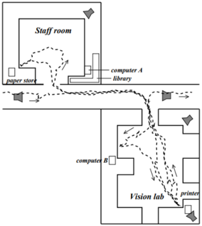
Plan Recognition as Policies

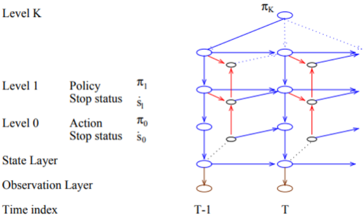
Various works that focus to the decision-making part in the agent's plan

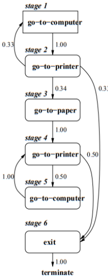
- Contingent plans
- Bayesian representations
- Game theory



- Defined Abstract Hidden Markov Memory Model
- MDP-based policies with memory (to remember what is the higher level goal in mind)
- Uses Particle Filter to observe current state
- In Figures: The environment; Two time slices as DBN; Memory transition for a policy

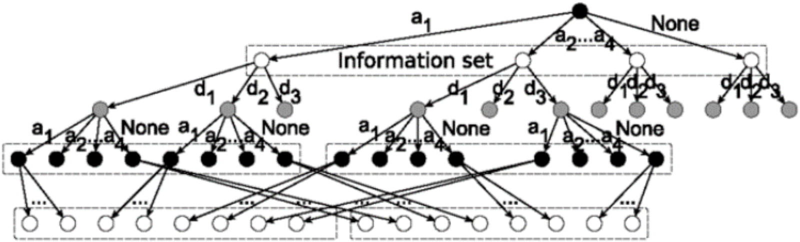






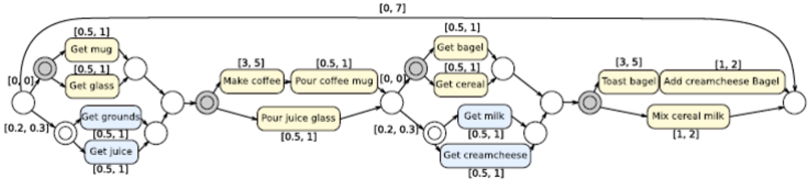
Game-Theoretic Approach to Adversarial Plan Recognition (Lisý et al., 2012)

- Translating the recognition problem to an imperfect-information extensive form game where
  - Actions are simultaneous
  - Actor has a set of actions to choose
  - Observer has a set of classes to choose
- With perfect knowledge, a solution is a Nash-equilibrium
- Without perfect knowledge, uses Monte-Carlo sampling



Concurrent Plan Recognition and Execution for Human-Robot Teams (Levine and Williams, 2014)

- Reasons about temporal constrains (e.g., breakfast must be made in 7 minutes)
- Temporal Plan Network under Uncertainty (TPNU)
- Actions are compiled into PDDL actions, and a solution is a PDDL-based plan that is consistent with time constraints
- Solved using Assumption-based Truth Maintenance System (ATMS)



Join the PAIR community

- Let's keep in touch: let us know if you have any questions:
  - **Chris:** cgeib@planrec.org
  - **Sarah:** sarah.e.keren@gmail.com skeren@seas.harvard.edu
  - **Reuth:** reuthde@gmail.com
- Come to the **PAIR workshop** tomorrow - we have 13 excellent papers and two invited talks.
  - Shlomo Zilberstein - Plan Recognition as a Multiagent Decision problem
  - David Smith - The Zoo of Interpretable Behavior
- Join our **mailing list:** plan-rec@googlegroups.com
- Explore our **website:** http://www.planrec.org/Resources.html
- Join us in organizing PAIR workshops