Dialogue Systems

Regina Barzilay

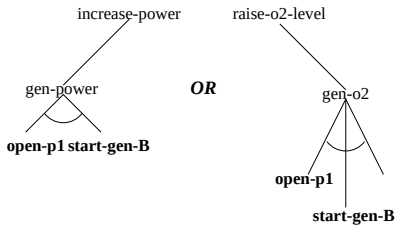
April 14, 2004

Plan Recognition

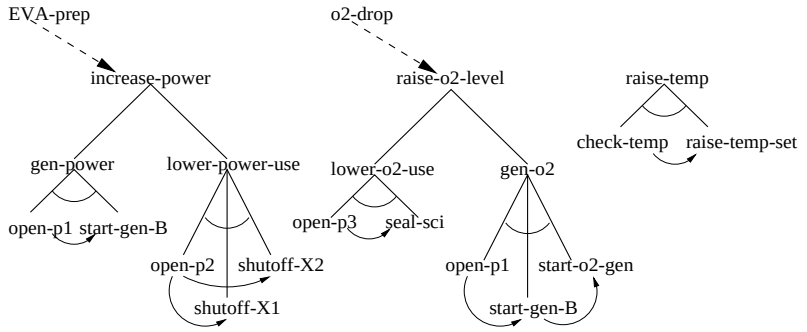
"Generalized Plan Recognition" (Kautz&Allen 1986)

- Plan recognition: lies in identification of a minimal set of *top-level actions* sufficient to explain the set of observed actions
- Plan representation: grap, with top level actions as root nodes and other actions as nodes depending from the top-level actions
- Approximation: the problem of graph recognition is a problem of graph covering

Example



Example



Plan Recognition: Algorithms

- (Charniak&McDermott,1985): plan recognition as abduction
  - (Charniak&Goldman,1993): use of Bayesian inference (supports minimal explanation, handles likelihood)
- (Vilain, 1990; Pynadath&Wellman, 1997): plan recognition as parsing (problems: does not support interleaved plans and partially-ordered plans)
- (Goldman&Geib&Miller, 1999): abductive, probabilistic plan recognition, centered around plan execution (supports interleaved plans, takes context into account)

Plan Recognition: Two views

Cohen, Perault and Allen (1981)

- Keyhole: the recognition is simply watching normal actions by an agent
- Intended: the agent is cooperative; its actions are done with the intent to be understood;

Notations

ACT - a set of actions

PRI $\in$ ACT

TOP $\in$ ACT (done for no other purposes than themsleves)

Primitive actions can be executed directly, while abstract actions are achieved indirectly by achieving other goals

Notations(cont.)

Plan (Beliefs concerning goal decomposition): (A, E, C)

A — set of actions

E — set of directed acyclic edges, $A_i \rightarrow A_j$ if A_j is a step in achieving A_i

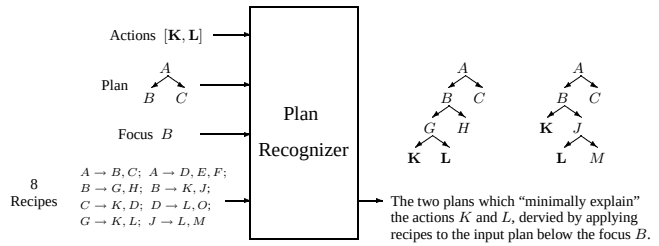
C — set of Constraints

Recipies: functions from non-primitive action A_i to (A', E', C'), such that actions in A' are steps towards A_i
Other functions: CONSISTENT?(P), REPLACE (P, A_i, A_j), DONE? (A)

Plan Recognition

- Input: $[A_1, \ldots A_n]$, P=(A, E, C), a focus of action f $\in$ P, recipy library
- Output: (possibly empty) set of extensions of P which "minimally explain" the input actions by applying recipies "below" the focus

Representation



Properties of Recognized Plans

- $[A_1, \ldots A_n] \in A'$
- every action in (A' - A) is reachable in E' from f
- P' can be derived from P by a composition of calls to EXTEND $(\ldots, R_k, \ldots)$, where $R_k \in R$, and replace$(\ldots, \ldots, A_k)$, where $A_k \in [A_1, \ldots A_n]$
- no smaller plan (A",E", C"), and A" $\in$ A', E" $\in$ E', C" $\in$ C'

Properties of Recognized Plans

- $[A_1, \ldots A_n] \in A'$
- every action in (A' - A) is reachable in E' from f
- P' can be derived from P by a composition of calls to EXTEND $(\ldots, R_k, \ldots)$, where $R_k \in R$, and replace$(\ldots, \ldots, A_k)$, where $A_k \in [A_1, \ldots A_n]$
- no smaller plan (A",E", C"), and A" $\in$ A', E" $\in$ E', C" $\in$ C'

Complexity

The size of the search space required to explain one action is bounded by: $F(R * S)^L$,
S is the maximum number of steps in a recipe
R is the maximum number of recipes applicable to the actions
F is the number of actions on the fringe of P
L is the lengthy of the longest sequence of recipes
$R_1, \ldots, R_L$ the algorithms must consider

Algorithm

(a) Plan recognition.

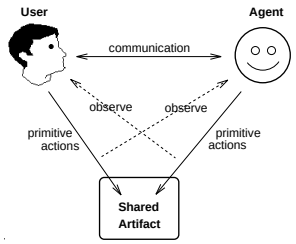
RECOGNIZE($[A_1, .., A_n]$,P,f,$\mathcal{R}$) ≡
$\mathcal{EXPL} \leftarrow \emptyset, \mathcal{Q} \leftarrow \emptyset$
if $P = \langle \emptyset, \emptyset, \emptyset \rangle$
  foreach $T_i \in \mathcal{TOP}$
    add $\langle [A_1, .., A_n], \langle \{T_i\}, \emptyset, \emptyset \rangle, T_i \rangle$ to $\mathcal{Q}$
else foreach $g_i \in$ FRINGE(P, f)
  add $\langle [A_1, .., A_n], P, g_i \rangle$ to $\mathcal{Q}$
until $\mathcal{Q} = \emptyset$
  remove $\langle [A'_1, .., A'_{n'}], P', act \rangle$ from $\mathcal{Q}$
  $P'' \leftarrow$ REPLACE(P', act, A'_1)
  if CONSISTENT?(P'')
    if $n' = 1$ add P'' to $\mathcal{EXPL}$
    else foreach $g_i \in$ FRINGE(P'', f)
      add $\langle [A'_2, .., A'_{n'}], P'', g_i \rangle$ to $\mathcal{Q}$
  if $act \notin \mathcal{PRIM}$
    foreach recipe $R_k \in \mathcal{R}$
      $P''' \leftarrow$ EXTEND(P', R_k, act)
      foreach s_j, where $act \rightarrow s_j \in P'''$
        add $\langle [A'_1, .., A'_{n'}], P''', s_j \rangle$ to $\mathcal{Q}$
return $\mathcal{EXPL}$

Algorithm(cont)

(b) Focus, ambiguity and clarification.

$plan \leftarrow \langle \emptyset, \emptyset, \emptyset \rangle$, $focus \leftarrow null$, $acts \leftarrow [\,]$
repeat
  wait for next input action A_i,
  if DONE?(root of $plan$)
    $plan \leftarrow \langle \emptyset, \emptyset, \emptyset \rangle$, $focus \leftarrow null$
  add A_i to $acts$
  pick $\leftarrow null$
  $\mathcal{EXPL} \leftarrow$ RECOGNIZE($acts, plan, focus, \mathcal{R}$)
  if $\mathcal{EXPL} = \emptyset$
    set $focus$ to root of $plan$
    $\mathcal{EXPL} \leftarrow$ RECOGNIZE($acts, plan, focus, \mathcal{R}$)
  if $|\mathcal{EXPL}| = 1$
    remove $pick$ from $\mathcal{EXPL}$
  else if $\mathcal{EXPL} = \emptyset$ or $|acts| > MaxWait$
    $pick \leftarrow$ CLARIFY($\mathcal{EXPL}$)
  if $pick \neq null$
    $plan \leftarrow pick$
    $focus \leftarrow$ UDATEFOCUS($plan, A_i$)
    $acts \leftarrow [\,]$

Plan recognition in Discourse



Example

without plan recognition

```
 1  USER:  "Let's work on email."
 2  AGENT: "You have 5 unread messages."
 3  USER:  "Let's read a message."
 4  USER:  <Select message from Chuck>
 5  USER:  "Let's read message in its own window."
 6  USER:  <Open message>
 7  USER:  "Let's do something using message."
 8  USER:  <Visit http://foo.bar.com>
 9         ...long lunch...
10  USER:  "What was I doing?"
11  AGENT: "Working on email.
12            Reading message from Chuck.
13             Reading message in its own window.
14              Done something using message.
15                <Visit http://foo.bar.com>
16            Expecting to react to message."
17  USER:  "Let's react to message."
18  USER:  "Let's send email."
19  USER:  <Start new message>
20  USER:  "Let's fill in message."
21  USER:  <Address message to Dan>
22
23
24
25
26         ...additional editing actions...
27  USER:  "Done filling in message."
28  AGENT: "Let's send, cancel or save msg."
29  USER:  "Please save message."
30  AGENT: <Save message>
```

with plan recognition

```
USER:  "Let's work on email."
AGENT: "You have 5 unread messages."

USER:  <Select message from Chuck>

USER:  <Open message>

USER:  <Visit http://foo.bar.com>
       ...long lunch...
USER:  "What was I doing?"
AGENT: "Working on email.
          Reading message from Chuck.
           Reading message in its own ...
            Done something using message.
              <Visit http://foo.bar.com>
          Expecting to react to message."


USER:  <Start new message>

USER:  <Address message to Dan>
AGENT: "Why you sending email to Dan?
        (a) reacting to msg from Chuck
        (b) as a new goal "
USER:  "I'm reacting to msg from Chuck."
       ...additional editing actions...
USER:  "Done filling in message."
AGENT: "Let's send, cancel or save msg."
USER:  "Please save message."
AGENT: <Save message>
```

Key: "utterance" <primitive action>