Planning: Example and Forward Planning

CPSC 322 Lecture 15

February 6, 2006
Textbook §11.1 - 11.2

Lecture Overview

Planning

Given:

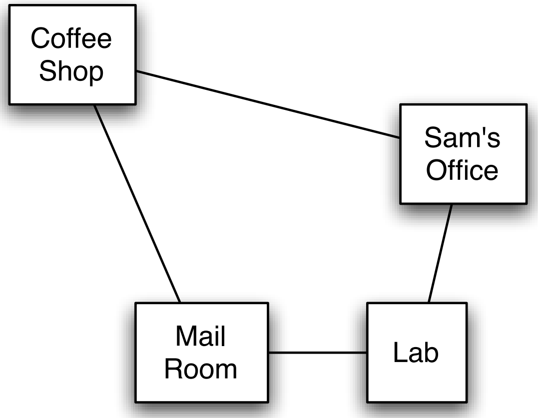
- A description of the effects and preconditions of the actions
- A description of the initial state
- A goal to achieve

find a sequence of actions that is possible and will result in a state satisfying the goal.

- recall: rather than looking for a single state that satisfies our constraints, look here for a sequence of states that gets us to a goal

Delivery Robot Example

Consider a delivery robot named Rob, who must navigate the following environment:



Delivery Robot Example

The state is defined by the following features:

$RLoc$ — Rob's location

- domain: coffee shop (cs), Sam's office (off), mail room (mr), or laboratory (lab)

RHC — Rob has coffee

- domain: true/false. By rhc indicate that Rob has coffee and by $\overline{rhc}$ that Rob doesn't have coffee.

SWC — Sam wants coffee (T/F)

MW — Mail is waiting (T/F)

RHM — Rob has mail (T/F)

An example state is $\langle lab, \overline{rhc}, swc, \overline{mw}, rhm \rangle$. How many states are there:

Delivery Robot Example

The state is defined by the following features:

$RLoc$ — Rob's location

- domain: coffee shop (cs), Sam's office (off), mail room (mr), or laboratory (lab)

RHC — Rob has coffee

- domain: true/false. By rhc indicate that Rob has coffee and by $\overline{rhc}$ that Rob doesn't have coffee.

SWC — Sam wants coffee (T/F)

MW — Mail is waiting (T/F)

RHM — Rob has mail (T/F)

An example state is $\langle lab, \overline{rhc}, swc, \overline{mw}, rhm \rangle$. How many states are there: $4 \times 2 \times 2 \times 2 \times 2 = 64$.

Delivery Robot Example

The robot's actions are:

$Move$ — Rob's move action

- move clockwise (mc), move anti-clockwise (mac) not move (nm)

PUC — Rob picks up coffee

- must be at the coffee shop

$DelC$ — Rob delivers coffee

- must be at the office, and must have coffee

PUM — Rob picks up mail

- must be in the mail room, and mail must be waiting

$DelM$ — Rob delivers mail

- must be at the office and have mail

Assume that Rob can perform one action of each kind in a single step. Thus, an example action is $\langle mc, \overline{puc}, \overline{dc}, \overline{pum}, dm \rangle$; we can abbreviate it as $\langle mc, dm \rangle$.

How many actions are there:

Delivery Robot Example

The robot's actions are:

$Move$ — Rob's move action

- move clockwise (mc), move anti-clockwise (mac) not move (nm)

PUC — Rob picks up coffee

- must be at the coffee shop

$DelC$ — Rob delivers coffee

- must be at the office, and must have coffee

PUM — Rob picks up mail

- must be in the mail room, and mail must be waiting

$DelM$ — Rob delivers mail

- must be at the office and have mail

Assume that Rob can perform one action of each kind in a single step. Thus, an example action is $\langle mc, \overline{puc}, \overline{dc}, \overline{pum}, dm \rangle$; we can abbreviate it as $\langle mc, dm \rangle$.

How many actions are there: $3 \times 2 \times 2 \times 2 \times 2 = 48$.

Lecture Overview

Planning

State-Based Rep

Feature-Based Rep

STRIPS

Forward Planning

State-Based Representation of a Planning Domain

- The domain is characterized by states, actions and goals
  - note: a given action may not be possible in all states
- Key issue: representing the way we transition from one state to another by taking actions
- We can't do better than a tabular representation:

Starting state	Action	Resulting state
⋮	⋮	⋮

- Problems with this representation:
  - too big
  - hard to modify
  - doesn't capture underlying structure

Example State-Based Representation

State	Action	Resulting State
$\langle lab, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle mc\rangle$	$\langle mr, \overline{rhc}, swc, \overline{mw}, rhm\rangle$
$\langle lab, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle mac\rangle$	$\langle off, \overline{rhc}, swc, \overline{mw}, rhm\rangle$
$\langle lab, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle nm\rangle$	$\langle lab, \overline{rhc}, swc, \overline{mw}, rhm\rangle$
$\langle off, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle mac, dm\rangle$	$\langle cs, \overline{rhc}, \overline{swc}, \overline{mw}, \overline{rhm}\rangle$
$\langle off, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle mac, \overline{dm}\rangle$	$\langle cs, \overline{rhc}, swc, \overline{mw}, rhm\rangle$
$\langle off, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle mc, dm\rangle$	$\langle lab, \overline{rhc}, \overline{swc}, \overline{mw}, \overline{rhm}\rangle$
$\langle off, \overline{rhc}, swc, \overline{mw}, rhm\rangle$	$\langle mc, \overline{dm}\rangle$	$\langle lab, \overline{rhc}, swc, \overline{mw}, rhm\rangle$
...	...	...

Lecture Overview

Planning

State-Based Rep

Feature-Based Rep

STRIPS

Forward Planning

Feature-Based Representation

- Represent states as joint assignments to a set of features rather than as black boxes
- Now we are looking for a sequence of variable assignments that
  - begins at the initial state
  - proceeds from one state to another by taking valid actions
  - ends up at a goal
- This means that instead of having one variable for every feature, we must instead have one variable for every feature at each time step, indicating the value taken by that feature at that time step!

Feature-Based Representation

We need two things to replace the tabular representation:

1. Modeling when actions are possible:
    - Provide a function that indicates when an action can be executed
    - Precondition of an action: a function (proposition) of the state variables that is true when the action can be carried out

Feature-Based Representation

We need two things to replace the tabular representation:

1. Modeling when actions are possible
2. Modeling state transitions in a "factored" way:
   - causal rules: explain how the value of a variable describing a feature at time step t depends on the action taken at time $t-1$
     - things that are changed in the world
     - example: $V_1 = v_1$ when act

Feature-Based Representation

We need two things to replace the tabular representation:

1. Modeling when actions are possible
2. Modeling state transitions in a “factored” way:
   - causal rules
   - frame rules: explain how the value of a variable describing a feature at time step t depends on the value of the variable that describes the same feature at time step $t-1$
     - things that are not changed in the world
     - example: $V_4 = v_7$ is maintained when condition c holds
     - the need for frame rules is counter-intuitive: but remember, a computer doesn’t know that the variables describing the same feature at different time steps have anything to do with each other, unless they are related to each other using appropriate constraints.

Example

When is $RLoc_{t+1} = cs$?

- $RLoc_{t+1} = cs$ when $RLoc_t = cs$ and $Move_t = nm$
- $RLoc_{t+1} = cs$ when $RLoc_t = off$ and $Move_t = mac$
- $RLoc_{t+1} = cs$ when $RLoc_t = mr$ and $Move_t = mc$

When is rhc true?

- $RHC_{t+1} = rhc$ when $RHC_t = rhc$ and $DelC_t = \overline{dc}$
- $RHC_{t+1} = rhc$ when $PUC_t = puc$

Which of these rules are frame rules?

Lecture Overview

Planning

State-Based Rep

Feature-Based Rep

STRIPS

Forward Planning

STRIPS

- The previous representation was feature-centric; STRIPS is action-centric.
- The STRIPS assumption:
  - all variables not explicitly changed by an action stay unchanged
- In STRIPS, an action has two parts:
  1. Precondition: a logical test about the features that must be true in order for the action to be legal
  2. Effects: a set of assignments to variables that are caused by the action

STRIPS vs. Feature Representation

- How can we write causal rules and frame rules for an action act with effects list $[V_1 = v_1, \ldots, V_k = v_k]$?

STRIPS vs. Feature Representation

- How can we write causal rules and frame rules for an action act with effects list $[V_1 = v_1, \ldots, V_k = v_k]$?
  - For each variable V_i in the effects list, write the causal rule "$V_i = v_i$ when act"
  - For each variable V_j not in the effects list, and every one of V_j's values v_k, write the frame axiom "$V_{j,t+1} = v_k$ when act_t and $V_{j,t} = v_k$"

Example

STRIPS representation of the action pick up coffee, PUC:

- **preconditions** $Loc = cs$ and $RHC = \overline{rhc}$
- **effects** $RHC = rhc$

STRIPS representation of the action deliver coffee, $DelC$:

- **preconditions** $Loc = off$ and $RHC = rhc$
- **effects** $RHC = \overline{rhc}$ and $SWC = \overline{swc}$

Note that Sam doesn't have to want coffee for Rob to deliver it; one way or another, Sam doesn't want coffee after delivery.

Lecture Overview

Planning

State-Based Rep

Feature-Based Rep

STRIPS

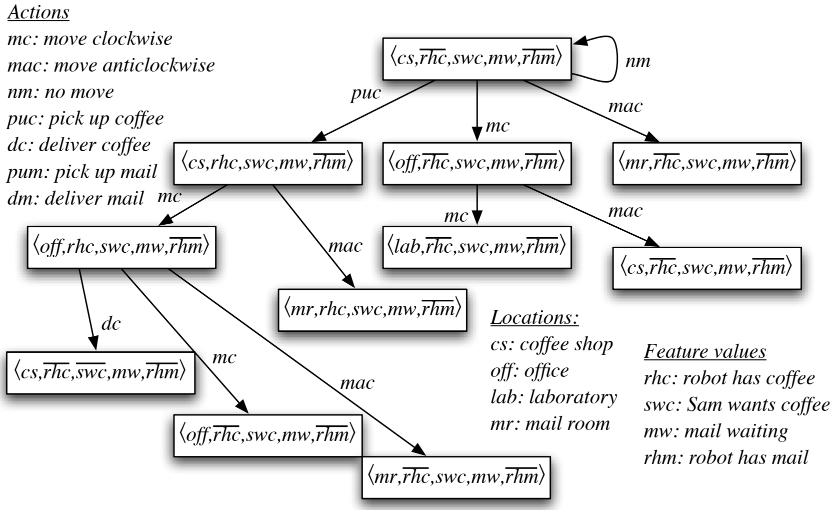
Forward Planning

Forward Planning

Idea: search in the state-space graph.

- The nodes represent the states
- The arcs correspond to the actions: The arcs from a state s represent all of the actions that are legal in state s.
- A plan is a path from the state representing the initial state to a state that satisfies the goal.

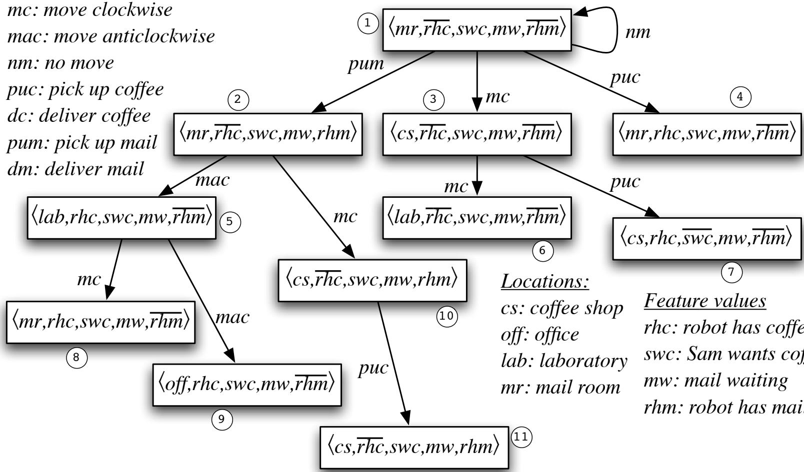
Example state-space graph



What are the errors (none involve room locations)?

Actions

mc: move clockwise
mac: move anticlockwise
nm: no move
puc: pick up coffee
dc: deliver coffee
pum: pick up mail
dm: deliver mail



Locations:

cs: coffee shop
off: office
lab: laboratory
mr: mail room

Feature values

rhc: robot has coffee
swc: Sam wants coffee
mw: mail waiting
rhm: robot has mail

Forward planning representation

- The search graph can be constructed on demand: thus, we only construct reachable states.
- If you want a cycle check or multiple path-pruning, you need to be able to find repeated states.
- There are a number of ways to represent states:
  - As a specification of the value of every feature
  - As a path from the start state

Improving Search Efficiency

Forward search can use domain-specific knowledge specified as:

- a heuristic function that estimates the number of steps to the goal
- domain-specific pruning of neighbors:
  - don't go to the coffee shop unless "Sam wants coffee" is part of the goal and Rob doesn't have coffee
  - don't pick-up coffee unless Sam wants coffee
  - unless the goal involves time constraints, don't do the "no move" action.