Planning: Representation and Forward Search

CPSC 322 Lecture 16

A Disney-inspired example

Consider the Disney/Pixar robot **Wall-E**

- What was his **purpose**?
- What **actions** could he take?
- Were there any **restrictions** on his actions?
- What kind of **plan** would he need to be able to make in order to carry out his function?

Learning Goals for today's class

You can:

- Represent a planning problem with the **STRIPS representation**
- Explain the **STRIPS assumption**
- Solve a planning problem by search (**forward planning**). Specify states, successor function, goal test and solution.

Lecture Overview

- **Where are we?**
- Planning
  - Example
  - STRIPS: a Feature-Based Representation
  - Forward Planning

R&R systems we'll cover in this course

		Environment	
Problem		**Deterministic**	**Stochastic**
Static	Constraint Satisfaction	*Variables + Constraints* Search Arc Consistency Local Search	
	Query	*Logics* Search	*Bayesian (Belief) Networks* Variable Elimination
Sequential	Planning	*STRIPS* Search	*Decision Networks* Variable Elimination

Representation
Reasoning Technique

Search for Specific R&R systems

Constraint Satisfaction Problems (note this is different for AC + domain splitting):

- State: assignments of values to a subset of the variables
- Successor function: assign values to a “free” variable
- Goal test: set of constraints
- Solution: possible world that satisfies the constraints
- Heuristic function: *none (all solutions at the same distance from start)*

Planning :

- State
- Successor function
- Goal test
- Solution
- Heuristic function

Inference

- State
- Successor function
- Goal test
- Solution
- Heuristic function

Lecture Overview

- **Where are we?**
- Planning
  - Example
  - STRIPS representation and assumption
  - Forward Planning

Planning as Search: State and Goal

How to select and organize a sequence of actions to achieve a given goal…

State: Agent is in a possible world (**full assignments** to a set of variables/features)

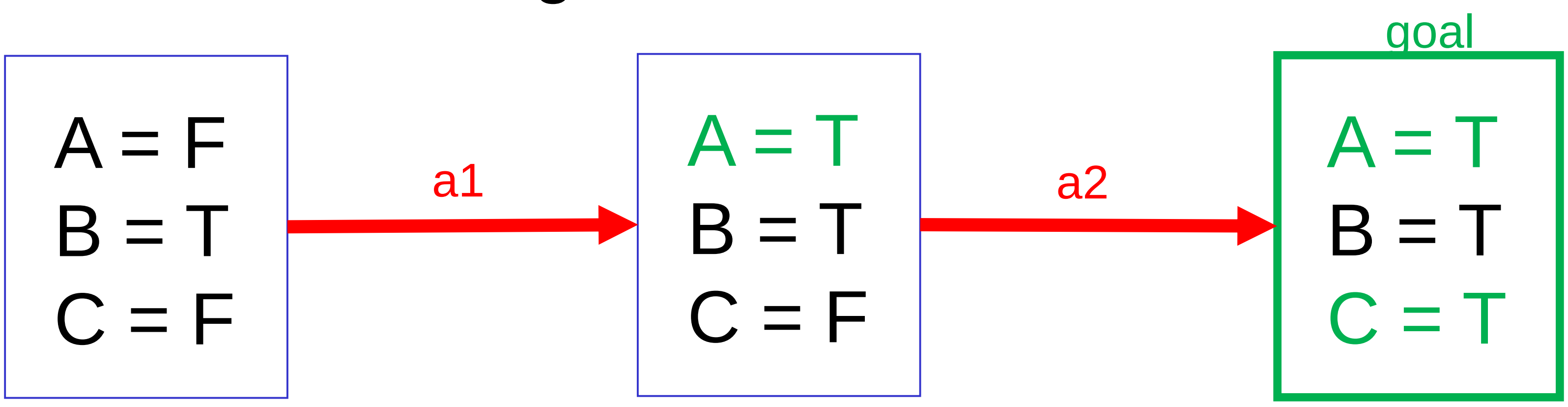
{A=F, B=T, C=F}

Goal: Agent wants to be in a possible world were **some** variables are given specific values

A=T, C=T

Planning as Search: Successor function and Solution

Actions : take the agent from one state to another



Solution: sequence of actions that when performed will take the agent from the current state to a goal state

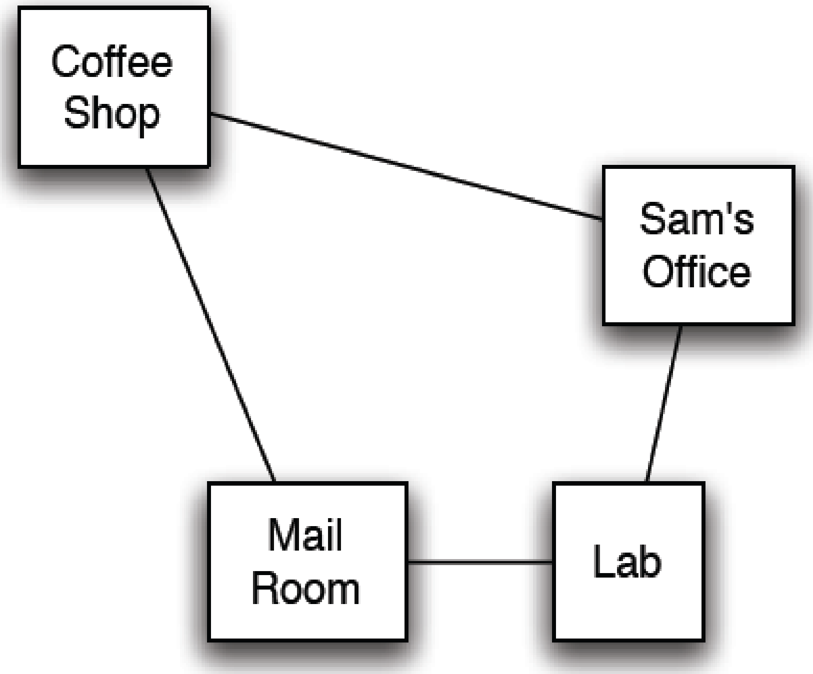
solution: (a1, a2)

Lecture Overview

- **Clarifications**
- **Where are we?**
- Planning
  - **Example**
  - STRIPS representation and assumption
  - Forward Planning

Delivery Robot Example (textbook)

Consider a **delivery robot named Rob**, who must navigate the following environment, can deliver coffee and mail to Sam



Another example will be available as a Practice Exercise: "Commuting to UBC"

Delivery Robot Example: States

The state is defined by the following variables/features:

RLoc - Rob's location

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

RHC - Rob has coffee **(T/F)**.

SWC - Sam wants coffee **(T/F)**

MW - Mail is waiting **(T/F)**

RHM - Rob has mail **(T/F)**

rhc	$\overline{rhc}$
T	F

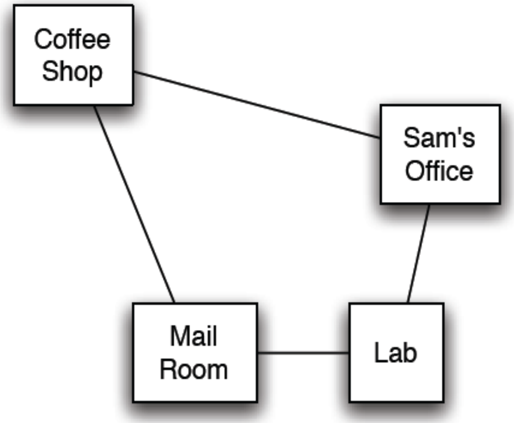
(alternate notation)

Example state: {cs, rhc, $\overline{\text{swc}}$, $\overline{\text{mw}}$, rhm}

Number of states:

A. 32 B. 64 C. 48 D. 16 E. 42

Delivery Robot Example: Actions



The robot's actions are:

Move - Rob's move action

- move clockwise (*mc*), move anti-clockwise (*mac*)

PUC - Rob picks up coffee

- must be at the coffee shop
- result: Rob has coffee

DelC - Rob delivers coffee

- must be at the office and must have coffee
- results: Rob does not have coffee, Sam does not want coffee

PUM - Rob picks up mail

- must be in the mail room, and mail must be waiting
- results: Rob has mail, mail is not waiting

DelM - Rob delivers mail

- must be at the office and have mail
- result: Rob does not have mail

preconditions
effects

Lecture Overview

- **Were are we?**
- Planning
  - Example
  - **STRIPS representation and assumption (STanford Research Institute Problem Solver)**
  - Forward Planning

STRIPS action representation

A key part of sophisticated planning is modeling actions

In STRIPS, an action has two parts:

1. Preconditions: a set of assignments to features that **must be satisfied** in order for the action to be legal
2. Effects: a set of assignments to features that are **caused** by the action

STRIPS actions: Example

STRIPS representation of the action pick up coffee, *PUC* :

- **preconditions** *Loc* = *cs* and *RHC* = F
- **effects** *RHC* = T

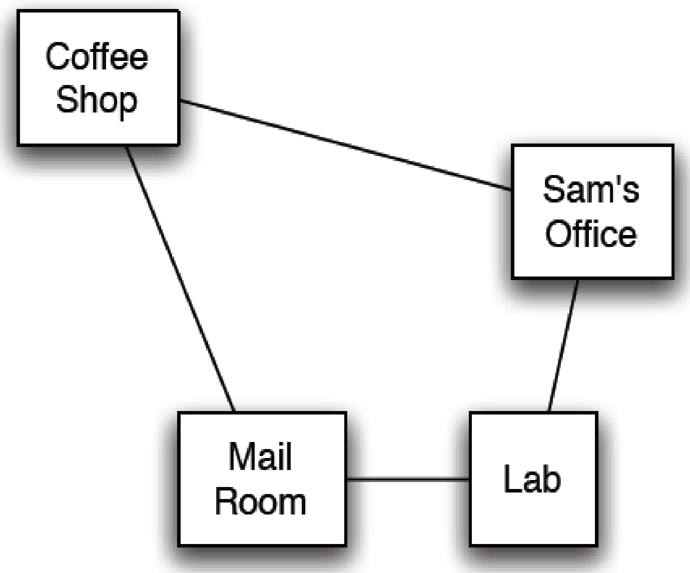
STRIPS representation of the action deliver coffee, *DelC* :

- **preconditions** *Loc* = *RHC* = (*SWC* =)
- **effects** *RHC* = *SWC* =

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

STRIPS actions: MC and MAC

STRIPS representation of the action Move Clockwise?



Precondition(s): *depends on location*

Effect(s): *depends on location*

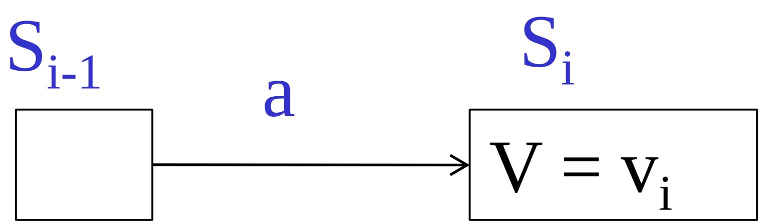
Need four different actions (one for moving clockwise from each location)

Need four more for moving counterclockwise (a.k.a. *anti-clockwise*)

STRIPS Actions (cont')

The STRIPS assumption: all features not explicitly changed by an action stay unchanged

What can we conclude about ***a*** and/or the state S_{i-1} below?



A. V was also v_i in S_{i-1}

B. One of the effects of **a** is to set $V = v_i$

C. A *OR* B

D. A *XOR* B

E. A *AND* B

Lecture Overview

- **Where are we?**
- Planning
  - Example
  - **STRIPS representation and assumption (STanford Research Institute Problem Solver)**
  - **Forward Planning**

Forward Planning

To find a plan (i.e. a solution): **search** in the state-space graph.

- The **states** are the **possible worlds**
- 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.

What actions **a** are legal/possible in a state **s**?

A. Those where **a**'s effects are satisfied in **s**

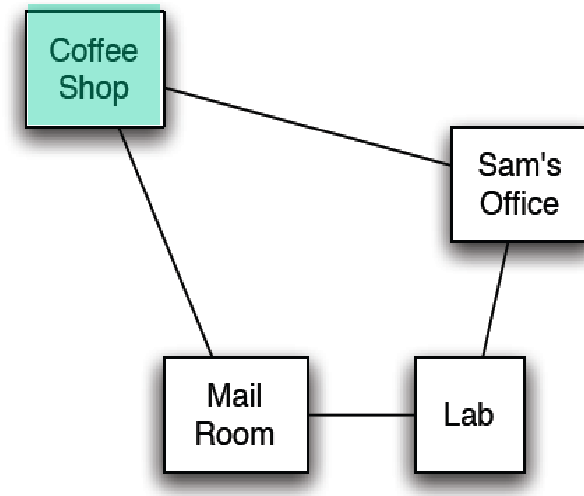
B. Those where **a**'s preconditions are satisfied in **s**

C. Those where the state **s'** reached via **a** is on the way to the goal

D. Those where **a**'s preconditions and effects are the same

E. How should I know, I'm not a lawyer

Example state-space graph: first level



recall there are different versions of each action based on location, but we'll simplify the notation here

Actions

mc: move clockwise

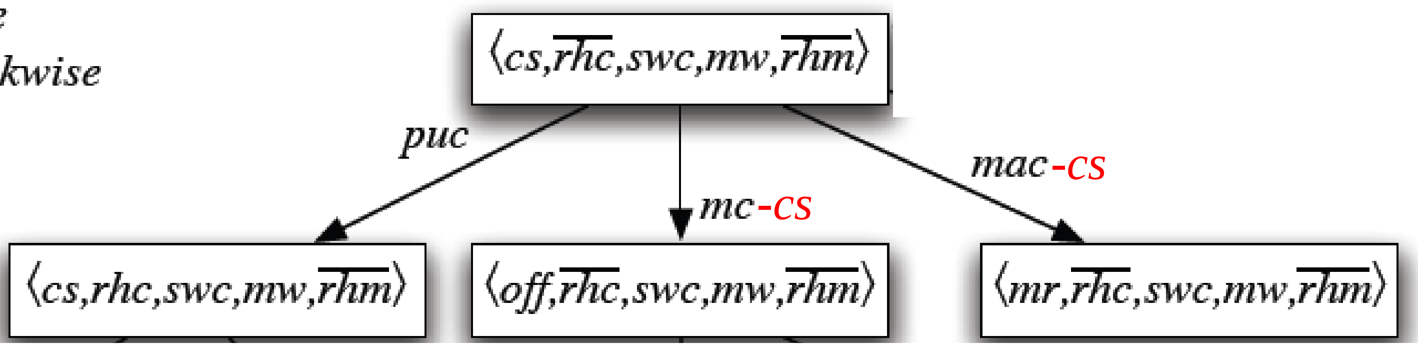
mac: move anticlockwise

puc: pick up coffee

dc: deliver coffee

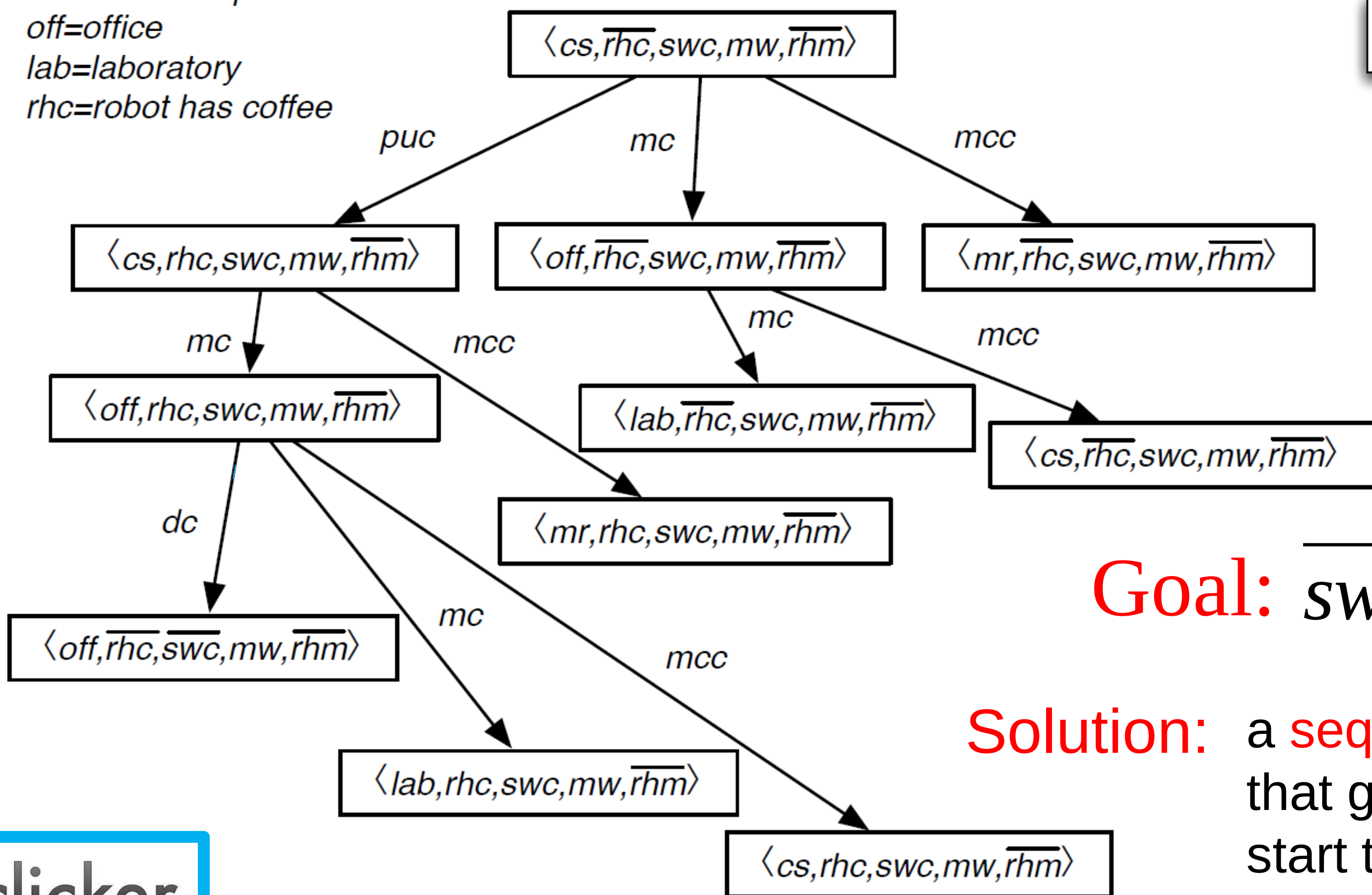
pum: pick up mail

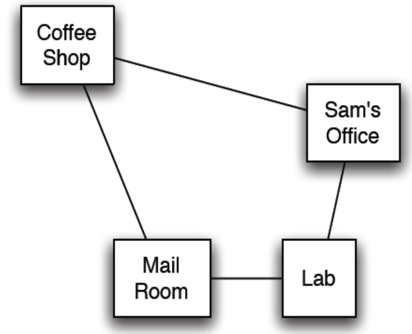
dm: deliver mail



Example for state space graph

cs=coffee shop
off=office
lab=laboratory
rhc=robot has coffee





Goal: $\overline{swc}$

Solution: a sequence of actions that gets us from the start to a goal

iclicker

Which is a solution to this planning problem?

A. (puc, mc)

B. (puc, mc, mc)

C. (puc, dc)

D. (puc, mc, dc)

Search for Specific R&R systems

Constraint Satisfaction Problems:

- State: assignments of values to a subset of the variables
- Successor function: assign values to a “free” variable
- Goal test: set of constraints
- Solution: possible world that satisfies the constraints
- Heuristic function: *none (all solutions at the same distance from start)*

Planning :

- State: full assignment of values to variables
- Successor function: actions with preconditions/effects
- Goal test: partial (or full) assignment of values to variables
- Solution: sequence of actions to goal
- Heuristic function: (next class)

Inference

- State
- Successor function
- Goal test
- Solution
- Heuristic function

Next:

Finish Planning (Ch. 6)

- Heuristics for planning (*not in textbook*)
- Mapping a planning problem into a CSP (6.4)