Planning: Representation and Forward Search

Computer Science cpsc322, Lecture 17

(Textbook Chpt 8.1 (Skip 8.1.1-2)- 8.2)

February, 11, 2009

Lecture Overview

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

Simulated Annealing: algorithm

Here's how it works (for maximizing):

- You are in node *n*. Pick a variable at random and a new value at random. You generate *n'*
- If it is an improvement i.e., $h(n') > h(n)$, adopt it.
- If it isn't an improvement, adopt it probabilistically depending on the difference and a temperature parameter, *T*.
  - we move to *n'* with probability $e^{(h(n')-h(n))/T}$

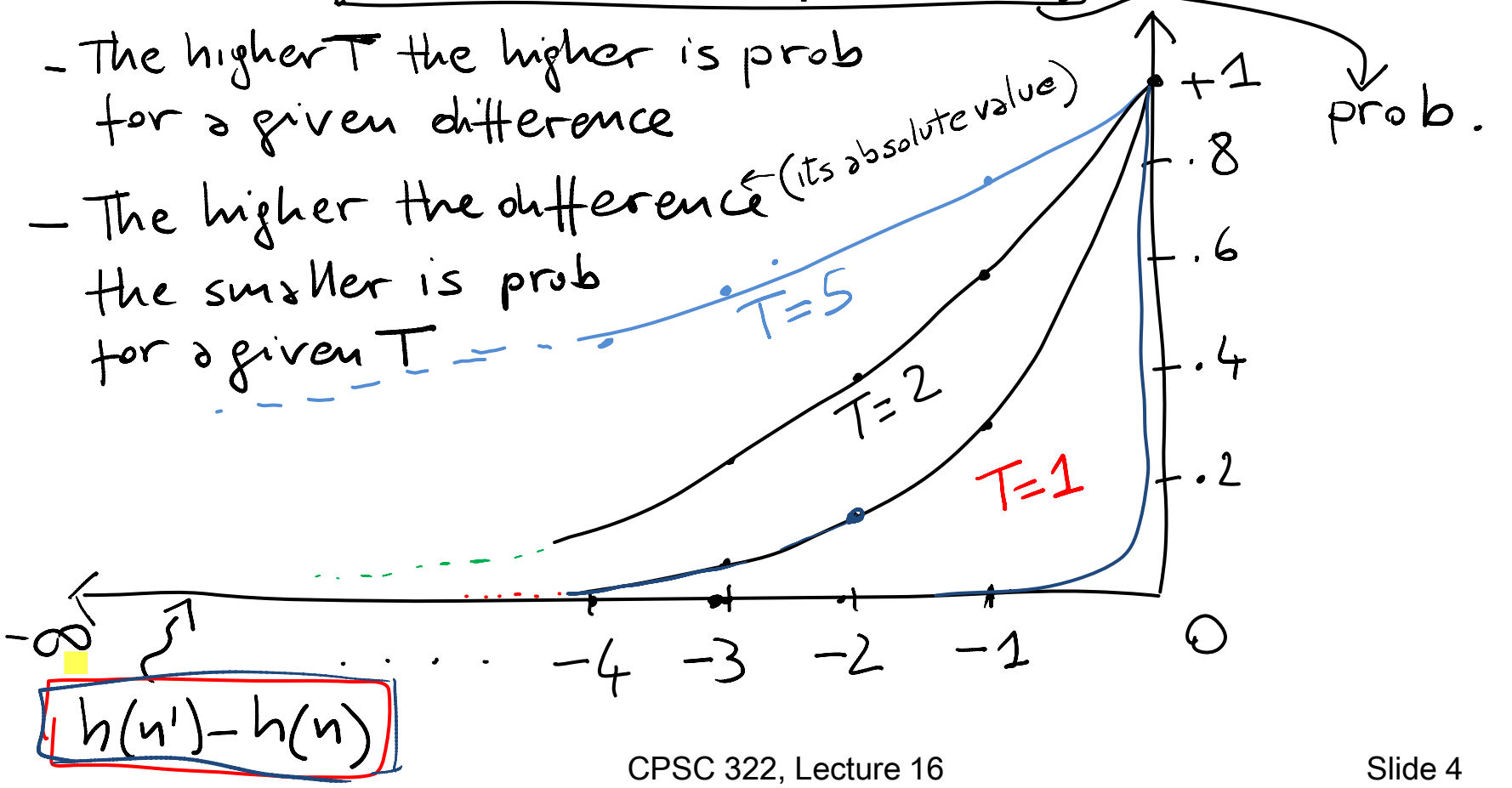
$h(n) > h(n') \Rightarrow h(n') - h(n) < 0$

see next slide

- If it isn't an improvement, adopt it probabilistically depending on the difference and a temperature parameter, *T*.

$h(n) > h(n') \Rightarrow h(n') - h(n) < 0$

  - we move to *n'* with probability $e^{(h(n')-h(n))/T}$

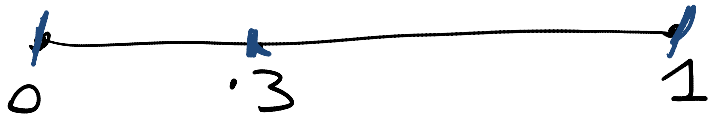


Sampling a discrete probability distribution

e.g. Sim. Annealing. Select n' with probability P

P = .3

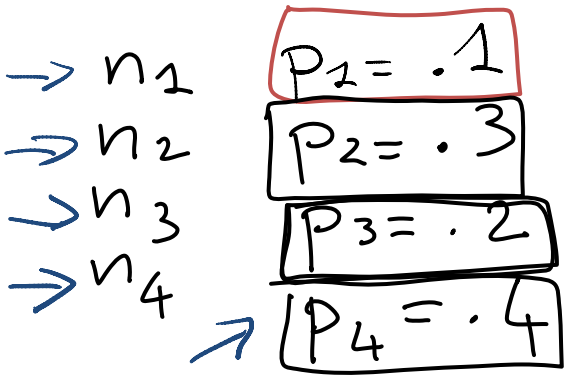
generate random number in [0,1]

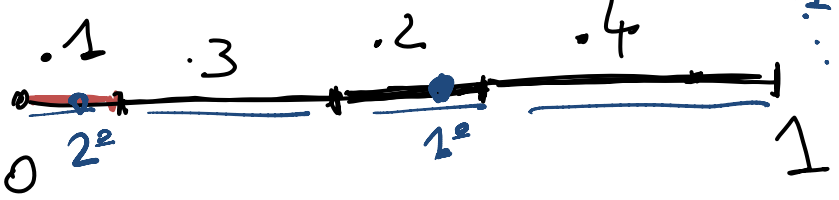


If < .3 accept n'

e.g. Beam Search: Select K individuals. Probability of selection proportional to their value

SAME HERE





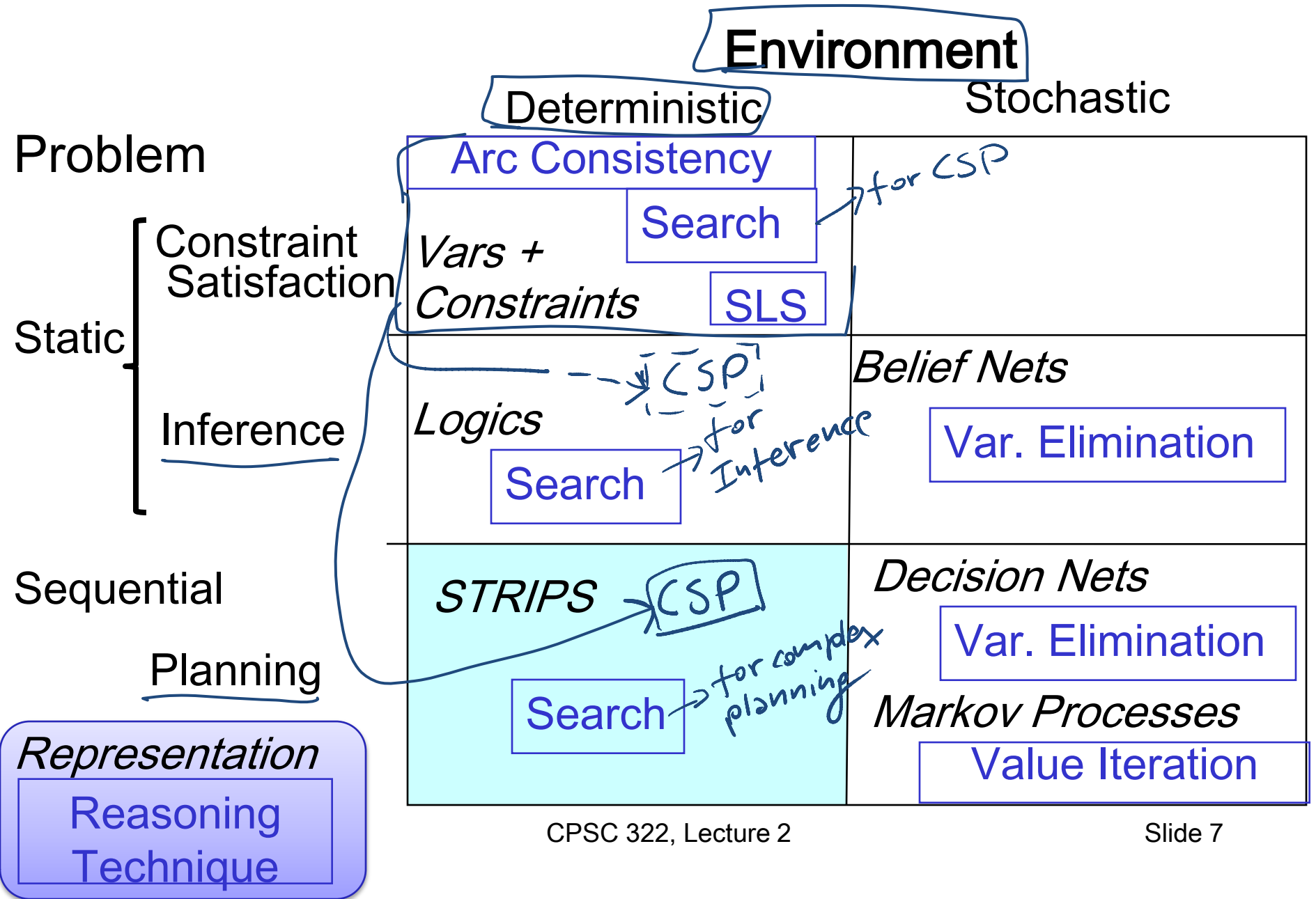
n_3 first sample

n_1 second sample

Lecture Overview

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

Modules we'll cover in this course: R&Rsys



Standard Search vs. 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 ✓
- Successor function ✓
- Goal test ✓
- Solution ✓
- Heuristic function next class

Inference

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

Lecture Overview

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

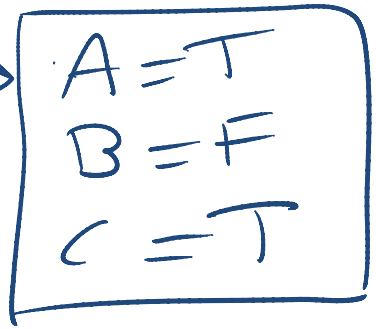
Planning as Search: State and Goal

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 B C

domain(true, false) (T, F) (1 0)

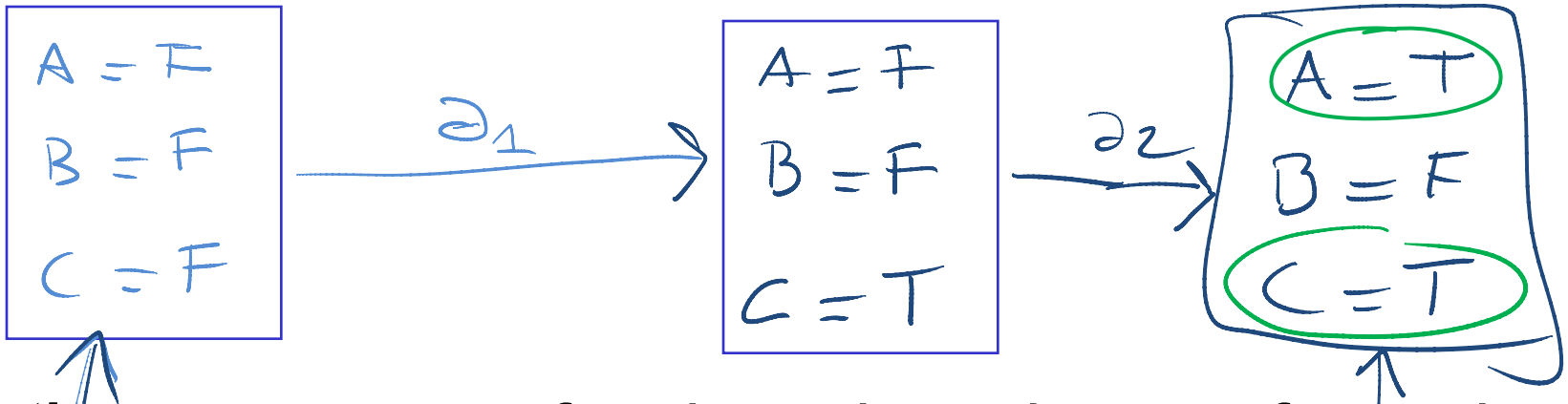


Goal: Agent wants to be in a possible world were some variables are given specific values

A=T and 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 (A = T and C = T) ← goal

start state

(a1 a2) SOLUTION/PLAN

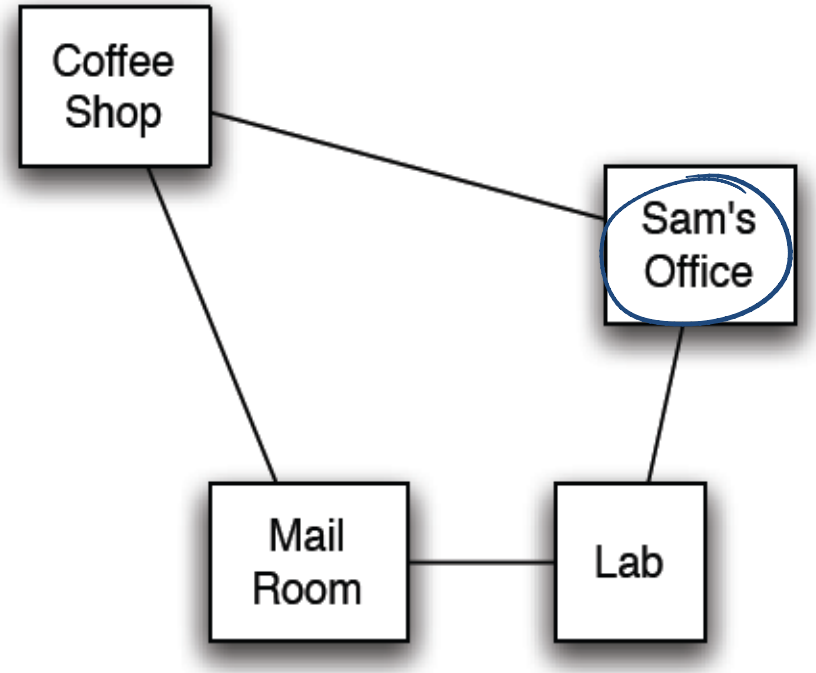
Goal

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



4 locations

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 True/False.

SWC - Sam wants coffee T/F

MW - Mail is waiting T/F

RHM - Rob has mail T/F

RHC=T
rhc
RHC

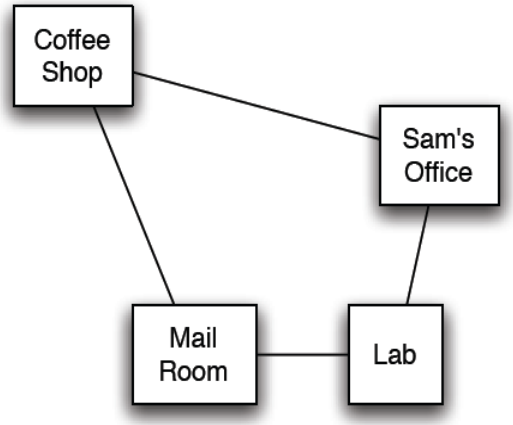
RHC=F
$\overline{rhc}$
¬RHC

alternative notations

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

Number of states: $2^4 \cdot 4 = 64$

Delivery Robot Example: Actions



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

preconditions

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

Lecture Overview

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

STRIPS action representation

The key to 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: Examples

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* = *off* and *RHC* = T (SWC = T)
- **effects** RHC = F and SWC = F

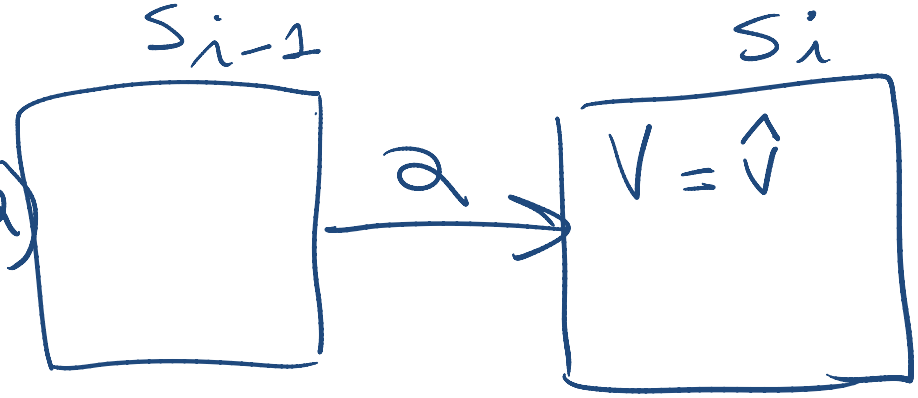
Note in this domain Sam doesn't have to want coffee for Rob to deliver it; one way or another, Sam doesn't want coffee after delivery.

STRIPS Actions (cont')

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

  - So if the feature/variable V has value $\hat{v}$ after the action a has been performed, what can we conclude about a and/or the state of the world immediately preceding the execution of a?

either
$V = \hat{v} \in effects(a)$
or $V = \hat{v}$ in s_{i-1}
or both



Lecture Overview

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

Forward Planning

To find a plan, a solution: search in the state-space graph.

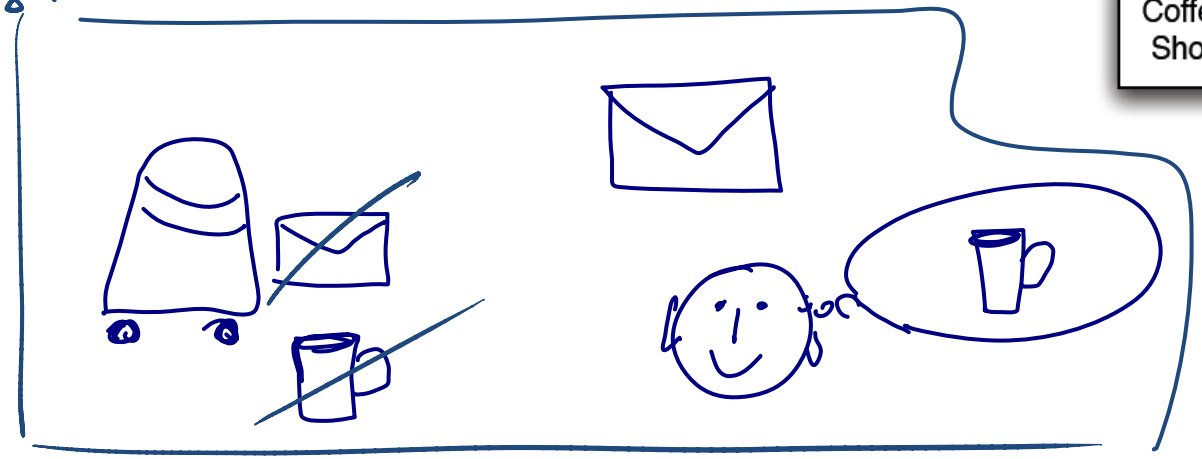
- The **states** are the **possible worlds**
- The **arcs** correspond to the **actions**: The arcs from a state *s* represent all of the actions that are legal in state *s*. *(What actions are legal?)*

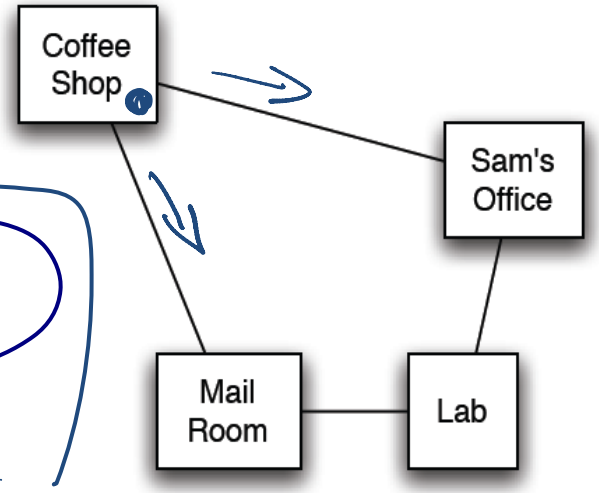
satisfy the preconditions

- A **plan** is a path from the state representing the initial state to a state that satisfies the goal.

Example state-space graph: first level

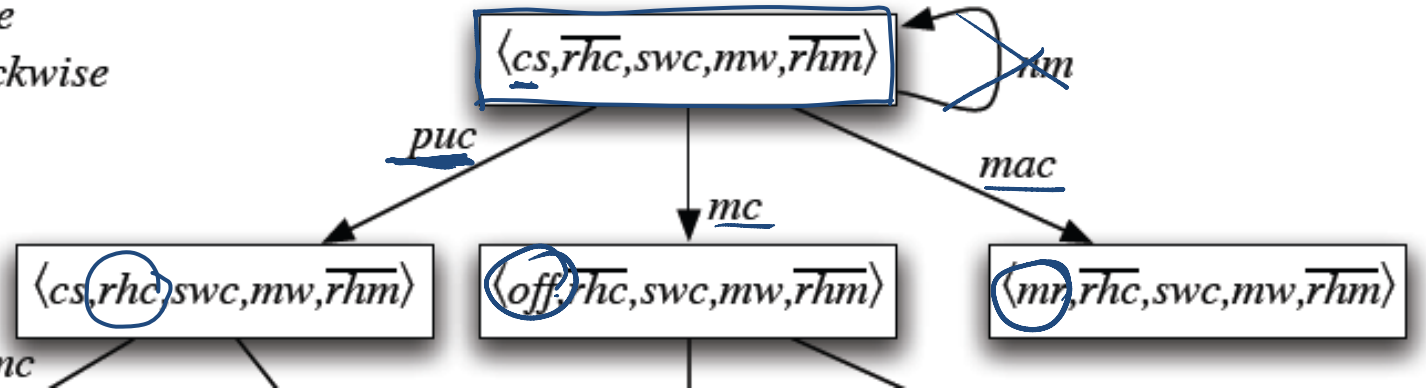
initial state





Actions

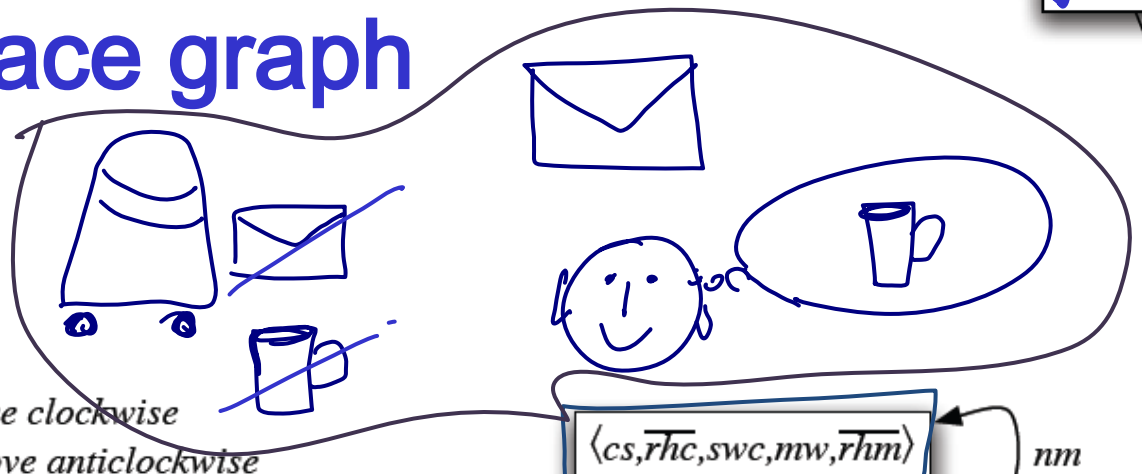
mc: move clockwise
mac: move anticlockwise
nm: no move
puc: pick up coffee
dc: deliver coffee
pum: pick up mail
dm: deliver mail

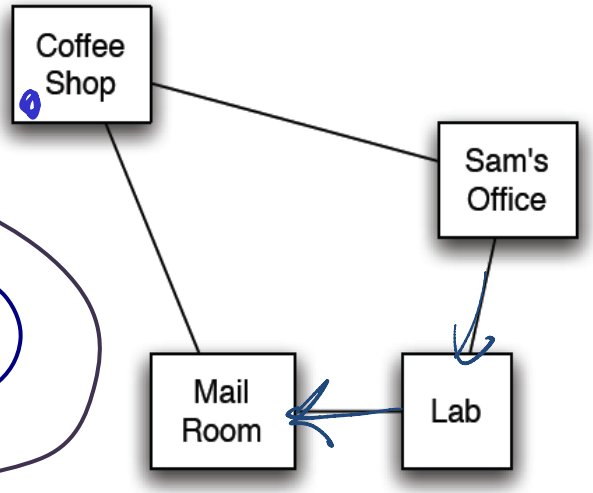


Goal $\overline{swc}$

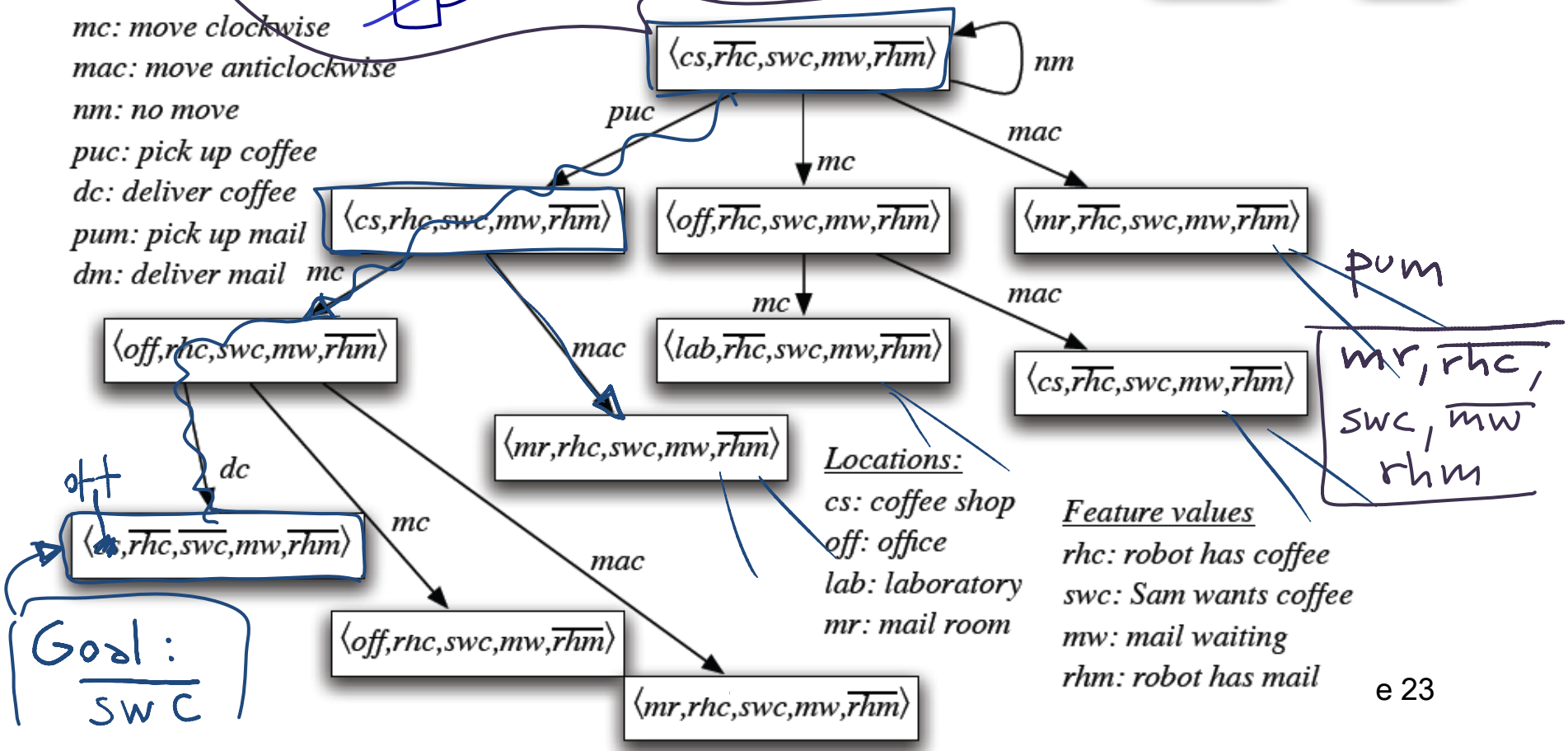
Example state-space graph

more levels





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

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.

Next class

Finish Planning (Chp 8)

- Heuristics for planning (not on textbook)
- Mapping planning problem into a CSP (8.4)