Planning
(Thu Apr 23)

- Slides based on Ch. 9 of Callan's *Artificial Intelligence*
- Outline
  - Planning & search
  - Strips language
  - Two approaches to planning problems
  - Exercises

Planning

- Planning = devising a sequence of actions acheiving a goal
- Essential for agents to act in an environment
- Growing research area, with a wide range of applications
  - NASA space mission planner
- Focus on **classical** planning
  - Fully observable; deterministic; finite; static; discrete
- Specific language to represent planning problems

Search & planning

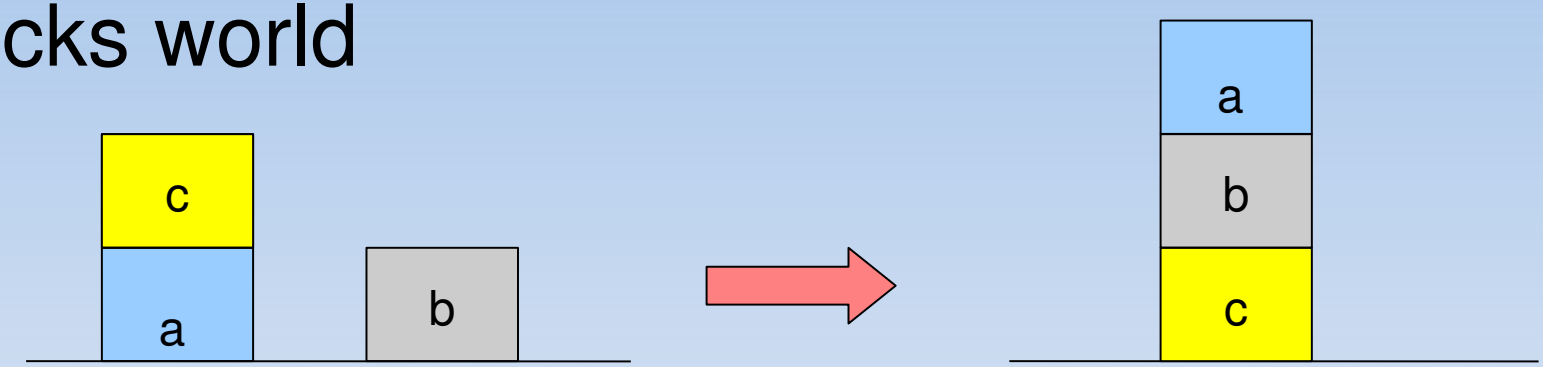
- Both concerned with finding a path from a start state to a goal
  - Plan = set of actions
- A planner uses more structural knowledge in deciding on its course of action:
  - What must be true in order for actions to be performed; effects of actions
- Interference of actions

Multi-agent planning

- In a multi-agent setting, the existence of correct joint plans does not mean the goal will be acheive
  - **coordination** to reach the same joint plan
- Not clear how a search procedure may help
- Example: the doubles tennis problem
- Ignore it in this lecture

A toy example

- Blocks world



- Constraints
  - Only one block can be moved at a time
  - One block can be moved onto another only if there is nothing on top of each

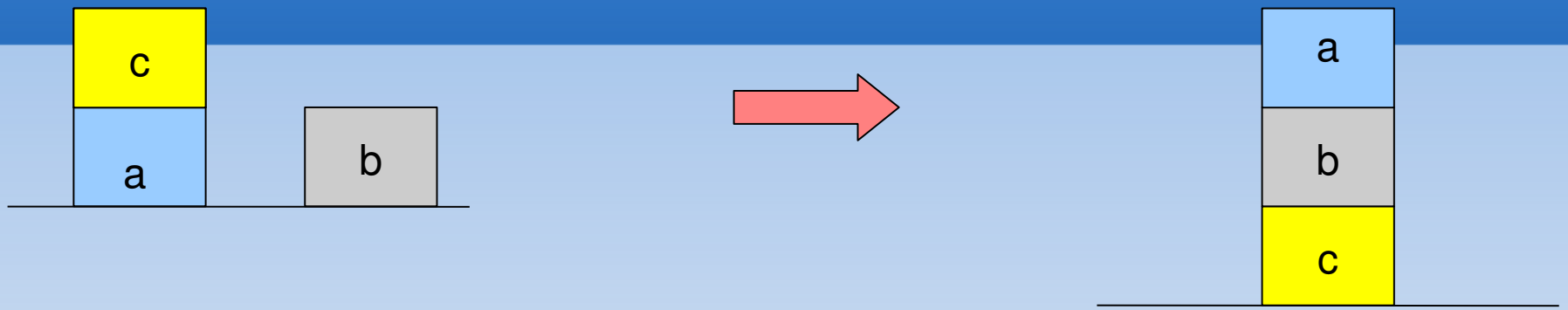
STRIPS language

- Offers a useful starting point to introduce the concepts of planning
- Used by many planners algorithms
- Developed in the 70s at the Stanford Research institute (Fikes and Nilsson, 1997)
- Three components
  - A description of the world
  - A description of the agent's goals
  - A description of the actions that an agent can perform

STRIPS

- Strips uses conjunctions of literals to describe the world and the task to be completed
- World state description
  - A conjunction of ground literals
- Goal description
  - A conjunction of literals

Blocks world (ctd)



Initial state description

on(a,fl) & on(c,a) & clear(c) & on (b,fl) & clear(b)

Goal description

clear(a) & on(a,b) & on(b,c) & on(c,fl)

- Only positive literals in states
  - Closed World Assumption: Unmentioned literals are false

Strips operators

- Used to represent actions
- Three core components
  - Action name and parameter list
  - Precondition = a conjunction of positive literals that specifies the conditions which must be true before the action is executed
  - Effect = a conjunction of both positive and negative literals that specify how the world changes after an action is executed
    - Add-list (for positive literals)
    - Delete-list (for negative literals)

Action description in Strips

- Action example: « *Move X from Y to Z!* »
- Strips operator

$$move(X,Y,Z)$$
$$precondition: on(X,Y) \wedge on(Y,Z) \wedge clear(Z)$$
$$add: on(X,Z) \wedge clear(Y)$$
$$delete: clear(Z) \wedge on(X,Y)$$

- Any variable in the precondition and the effect list must also appear in the parameter list
- A strip operator may have many instanciations
- Note the use of the 'clear' conjunct

A typical exam question

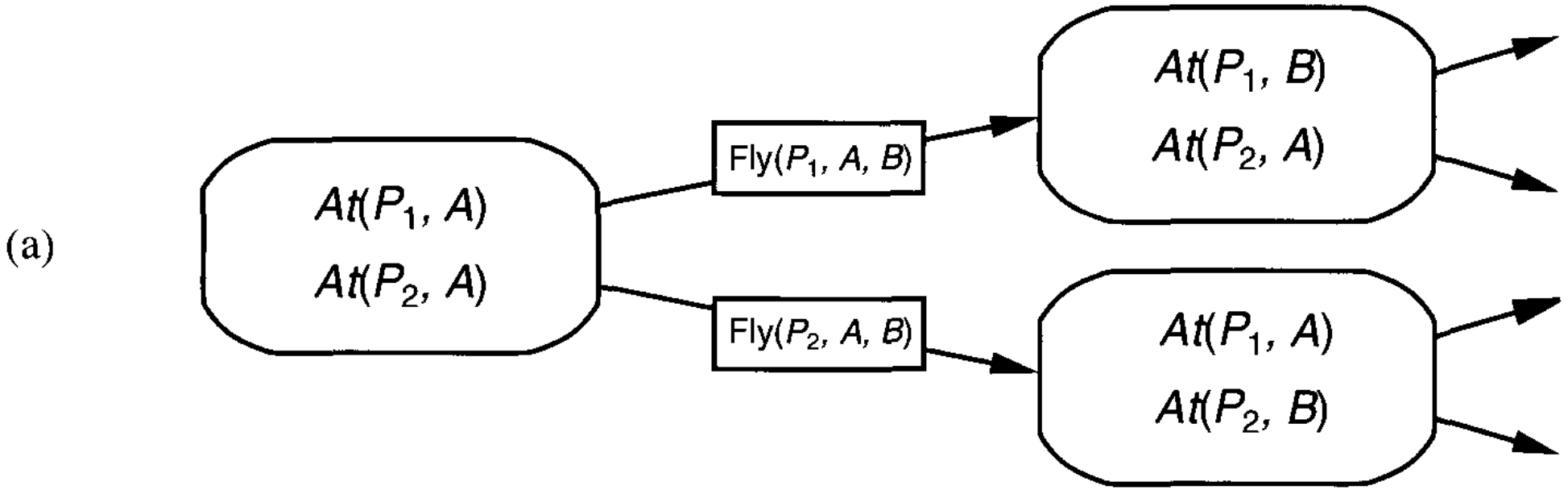
- Write suitable operators for such and such action. What does « suitable » mean in this context?
- Sample exercise: the towers of Hanoi

Planning as search

- Searching a state space
  - Each node in the graph denotes a state of the world. Arcs in the graph correspond to the execution of a specific action.
- Planning problem
  - Find a path from the initial state to the goal state.
    - sequence of actions
- Naive method is forward looking
  - Progression planning

Progression planning

- Generate all the actions that can be performed in a given state, starting from the initial state



- Check whether a given state is a goal state
  - search algorithm: breadth-first, depth-first, etc

Sussman anomaly

- Sub-optimal solutions picked up
  - Do and undo actions unnecessarily
- Illustration: Blocks-world scenario
  - on(a,b) and on(b,c) treated separately

Start State

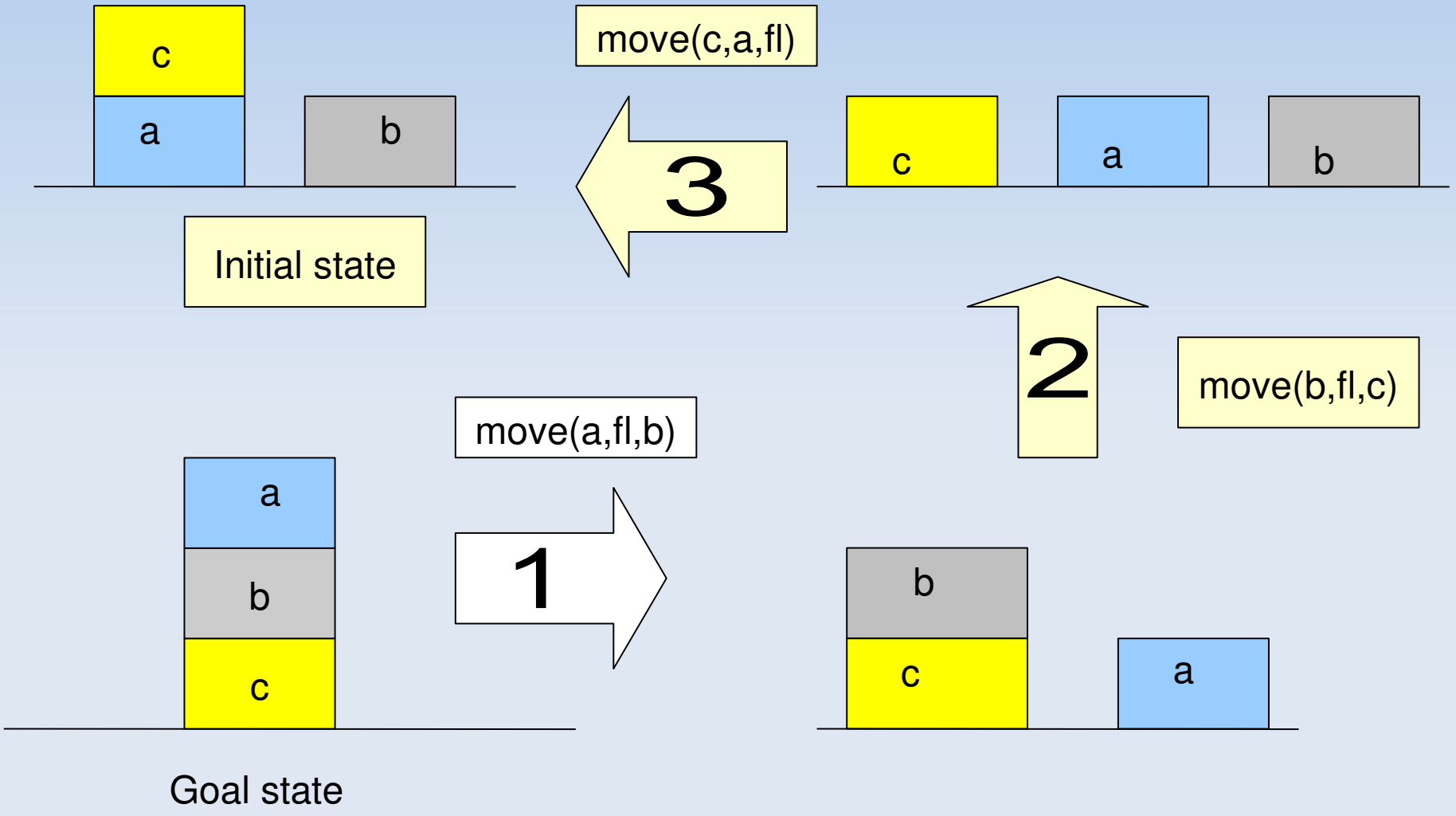
Goal State

Regression planning

- Start from the goal state
  - Pick up actions whose effects match one (or more) of the sub-goals, and
  - Post the chosen action's preconditions as new sub-goals (goal regression)
  - Continue until the start state is reached
- Only relevant actions are considered

Regression planning

- Blocks scenario



Regression planning

- Not exactly the same as working forward from the finish state until the start state is found
- To see why, just compare the sequence of actions obtained using each method
  - move(a, fl,b) becomes move(a, b, fl)
    - Likewise for the other actions involved
  - Using the second method, the sequence of actions is no solution unless further manipulation is done
- Another illustration: one-way rocket problem
  - Load (cargo) becomes unload(cargo)

Progression vs Regression

- This is a search rather than a planning problem
- Progression is data-driven. May do lots of work that is irrelevant to the goal
  - e.g., natural deduction
- Regression is goal-driven, appropriate for problem-solving
  - e.g., semantic tableaux
- Complexity of regression planning can be much less than that of progression planning

Exercise

- Show the steps in regression planning when searching for a solution to the blocks problem, taking the initial state to be as before and the goal state to be