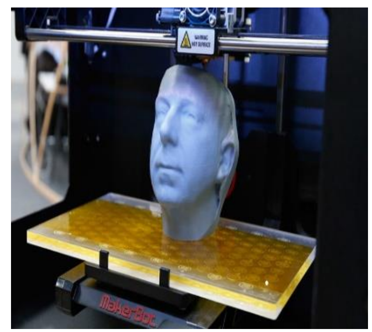
Automatic Planning

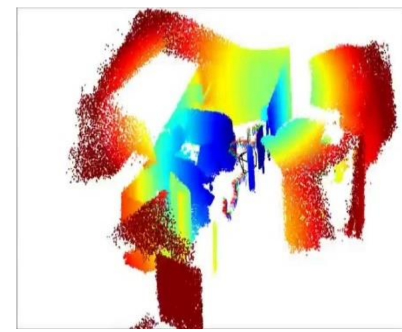
Hatem A. Rashwan

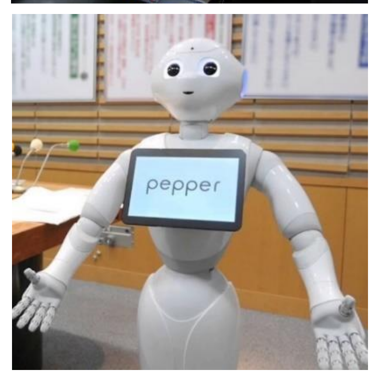
Introduction and Planning Definition

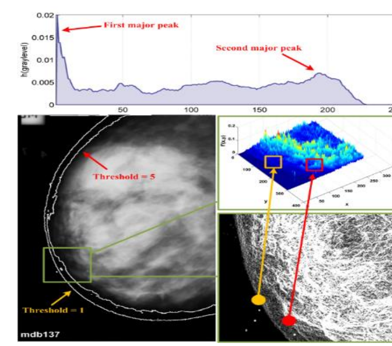
About the instructor

- **My name is Hatem A. Rashwan**
- **Research Group**: http://deim.urv.cat/~rivi/
  - The IRCV group is constituted by faculty from the Department of Computer Science and Mathematics (DEIM) and the Department of Electrical, Electronic and Automation Engineering (DEEEA). Both departments are physically located at the School of Engineering (ETSE) in Tarragona(Catalonia-Spain).
- **Research Interests:**
  - Computer Vision,
  - Pattern Recognition
  - Machine/Deep Learning
  - Artificial Intelligence
- **Applications**:
  - Vision-based robotic systems
  - Scene understanding
  - Productivity









Some of the topics covered in the PAR course

- Introduction and Planning definition
- State-Space Search: Heuristic Search and STRIPS
- Plan-Space Search and Hierarchical Task Network (HTN) Planning
- Graphplan and Advanced Heuristics
- Plan Execution and Applications
- Mobile Robot Application

Books covered the topics

Books

- Automated planning theory and practice, http://homes.dcc.ufba.br/~thiagob052/AI%20Planning/livro-recomendado.pdf **Chapters 1,2,4,6,9, and 20**
- Automated planning and acting, http://projects.laas.fr/planning/book.pdf, **Chapters 6 and 7**

Books for PDDL

- An Introduction to the Planning Domain Definition Language (PDDL), https://courses.cs.washington.edu/courses/cse473/06sp/pddl.pdf, http://homepages.inf.ed.ac.uk/mfourman/tools/propplan/pddl.pdf

Planning software

For Windows:

- Visual studio Code with PDDL packages, https://marketplace.visualstudio.com/items?itemName=jan-dolejsi.pddl

For Linux:

- Visual studio Code with PDDL packages,
- FF (Fast-Forward) Planning Software: http://www.ai.mit.edu/courses/16.412J/ff.html
- Graphplan Planning Software: http://www.ai.mit.edu/courses/16.412J/Graphplan.html

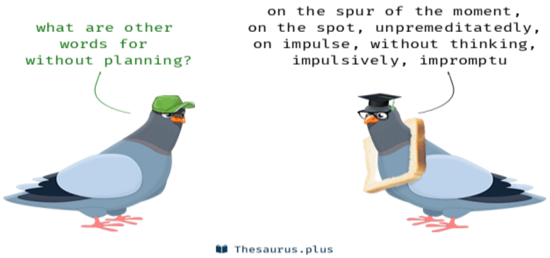
Online PDDL Editor: http://editor.planning.domains/

Overview

- **What is Planning (in AI)?**
- A Conceptual Model for Planning
- Planning and Search
- Example Problems

Human Planning and Acting

- ❑ Acting without (explicit) planning:
  - when purpose is immediate
  - when performing well-trained behaviours
  - when course of action can be freely adapted



- ❑ Acting after planning:
  - when addressing a new situation
  - when tasks are complex
  - when the environment imposes high risk/cost
  - when collaborating with others

- ❑ people plan only when strictly necessary

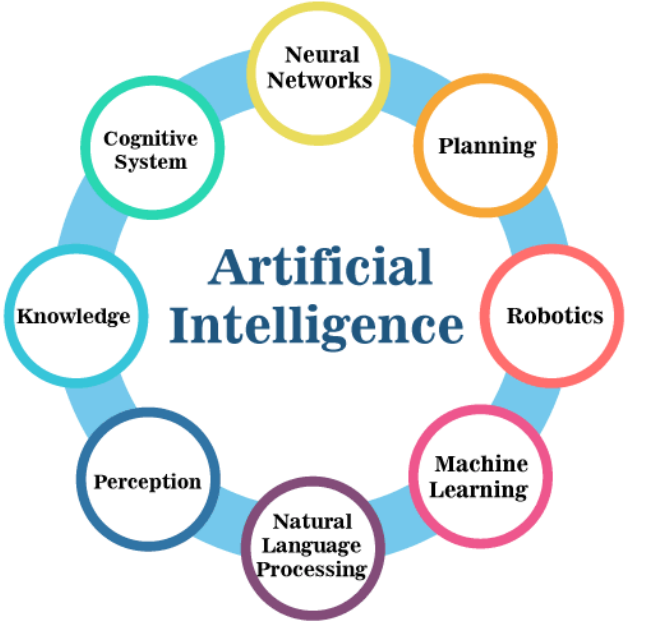
What is Planning?

- ❑ Planning:
  - explicit deliberation process that chooses and organizes actions by anticipating their outcomes
  - aims at achieving some pre-stated objectives

❑ AI planning: computational study of this deliberation process

Why Study Planning in AI?

- Scientific goal: understand intelligence
  - planning is an important component of rational (intelligent) behaviour
- Engineering goal: build intelligent entities
  - build planning software for choosing and organizing actions for autonomous intelligent machines



Attributes of Planning

Planning has seven attributes:

- Managerial function
- Goal oriented
- Pervasive
- Continuous Process
- Intellectual Process
- Futuristic
- Decision making

Domain-Specific vs. Domain-Independent Planning

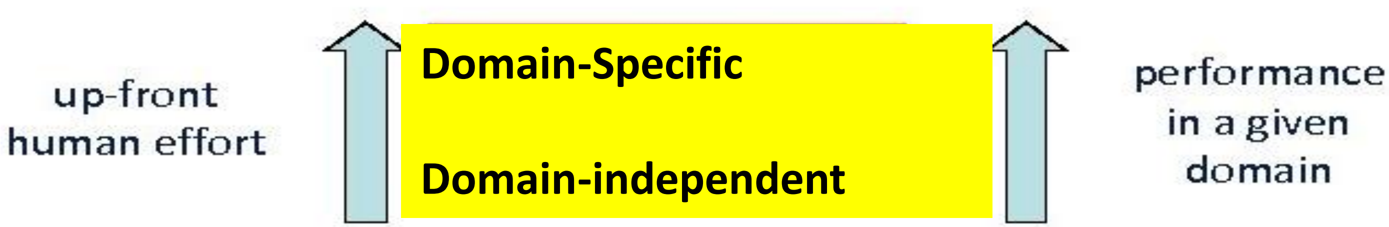
- **Domain-specific planning**: use specific representations and techniques adapted to each problem
  - important domains: path and motion planning, perception planning, manipulation planning
  - Do not work well in other planning domains

- **Domain-independent planning**: use generic representations and techniques
  - exploit commonalities to all forms of planning
  - leads to general understanding of planning

- **Domain-independent planning complements domain-specific planning**

Domain-Specific vs.
Domain-Independent Planning

Comparisons



- Domain-specific planner
  - Write an entire computer program - lots of work
  - Lots of domain-specific performance improvements
- Domain-independent planner
  - Just give it the basic actions - not much effort
  - Not very efficient

Remind

True or False?

- People only plan *when they have to* because the benefit of an optimal plan does not always justify the effort of planning
- For humans, planning is a *subconscious process*, which is why computational planning is so hard
- Planning involves a *mental simulation of actions* to foresee future world states and compare them to goals
- In AI, planning is concerned with the search for *computationally optimal* plans
- Domain-specific planning is used when efficiency is vital, whereas domain-independent planning is good for planning from first principles

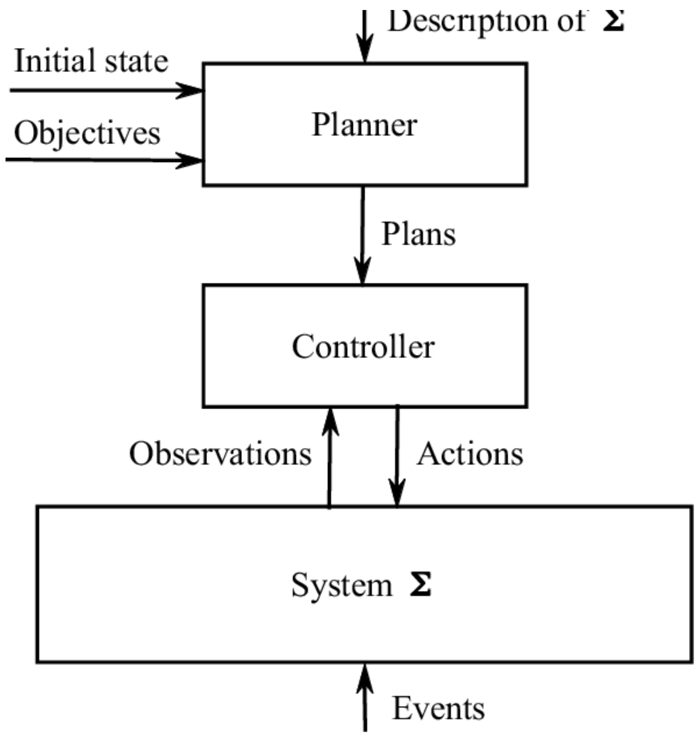
Overview

- What is Planning (in AI)?
- **A Conceptual Model for Planning**
- Planning and Search
- Example Problems

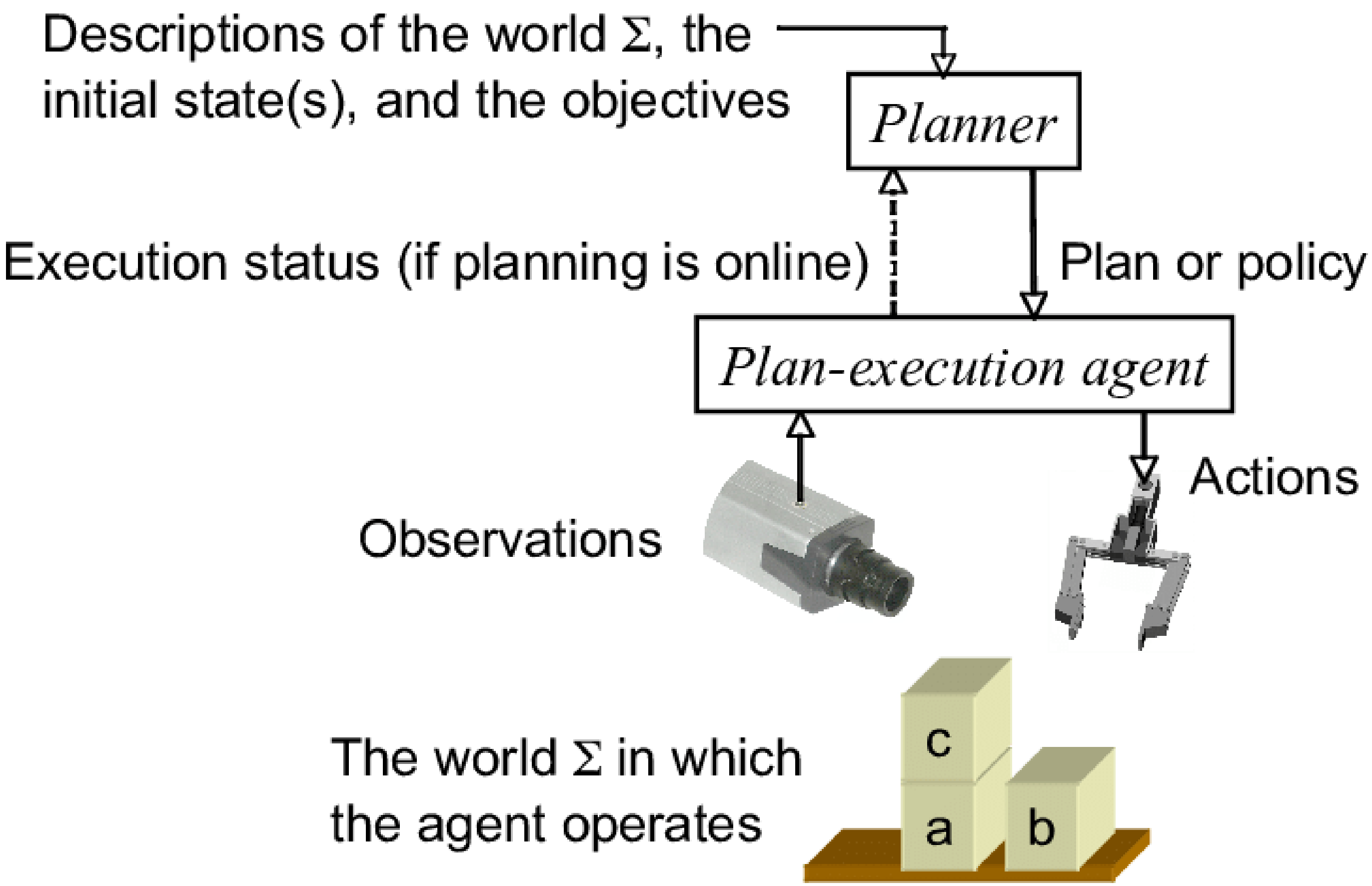
Conceptual Model?

A conceptual model is **a representation of a system** that consists of concepts used to help agents **know, understand, or simulate** a subject that the model represents.

- conceptual model: theoretical method for describing the elements of a problem
- good for:
  - explaining basic concepts
  - clarifying assumptions
  - analysing requirements
  - proving semantic properties (optimal, sound or complete)
- not good for:
  - efficient algorithms and computational concerns



Conceptual Model?



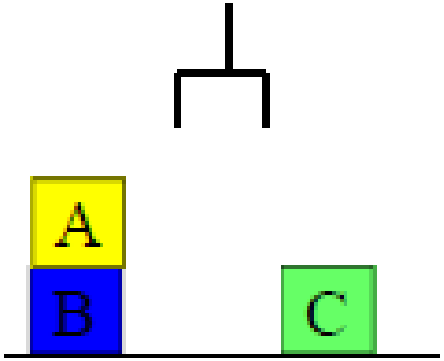
Conceptual Model for Planning: State-Transition System

- A state-transition system is a triple $\Sigma = (S,A,\gamma)$, where:
  - $S = \{s_1,s_2,\ldots\}$ is a finite or recursively enumerable set of states;
  - $A = \{a_1,a_2,\ldots\}$ is a finite or recursively enumerable set of actions;
  - γ: $S \times A \rightarrow 2^S$ is a state transition function.
- if $a \in A$ and $\gamma(s,a) \neq \emptyset$ then a is applicable in s
- applying a in s will take the system to $s' \in \gamma(s,a)$

State-Transition System
Blocks World Problem

➢ What is a State and Goal?

- We'll illustrate the techniques with reference to the blocks world
- This world contains
  - a robot arm with gripper,
  - 3 blocks (A, B and C) of equal size,
  - a table-top.
- Some domain constraints:
  - Any number of blocks can be directly on top of another block
  - Any number of blocks can be on the table
  - The hand can only hold one block

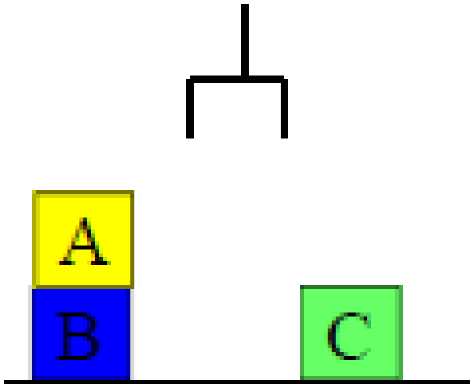


The Blocks World
States declaration

➢ What is a State and Goal?

To represent this environment, we need an Ontology (logical)

- ❑ **On(x,y)** *means block x* is on top of *block y*
- ❑ **OnTable(x)** --- *block x* is on the table
- ❑ **Clear(x)** --- nothing is on top of *block x*
- ❑ **Holding(x)** --- *robot* arm is holding *block x*
- ❑ **ArmEmpty()** --- *robot* arm/hand is not holding anything (block in this world)

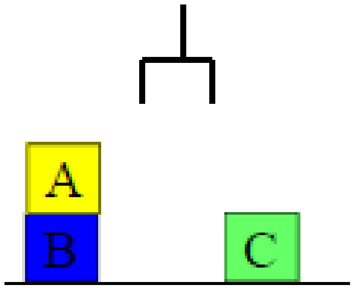


Blocks World State and Goal Description
States and Goal Representation

➢ State Representation = Environment

- A representation of one state of the blocks world.

  The state in the figure is:
  - *Clear(A)*
  - *Clear(C)*
  - *On(A,B)*
  - *OnTable(B)*
  - *OnTable(C)*
  - *ArmEmpty()*



- Use the *closed world assumption*: *anything not stated is assumed to be false*

➢ Goal Representation

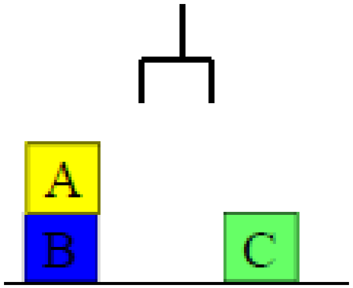
- A *goal* is represented as a set of formulae. Here is a goal:

  OnTable(A)
  OnTable(B)
  OnTable(C)

Blocks World
Actions Description

How do we can define an Action?

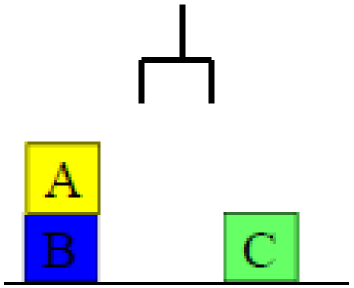
- Represented using a technique that was developed in the STRIPS planner. Each action has:
  - a **name** **---**which may have arguments;
  - a **pre-condition list** --- a list of facts which must be true for action to be executed;
  - a **delete list** --- a list of facts that are no longer true after action is performed;
  - an **add list** --- a list of facts made true by executing the action.
  - Each of the facts may contain **variables**



Blocks World Actions Description

Action/Operator Representation

- Basic operations
  - stack(X,Y): put block X on block Y
  - unstack(X,Y): remove block X from block Y
  - pickup(X): pickup block X from the table
  - putdown(X): put block X on the table

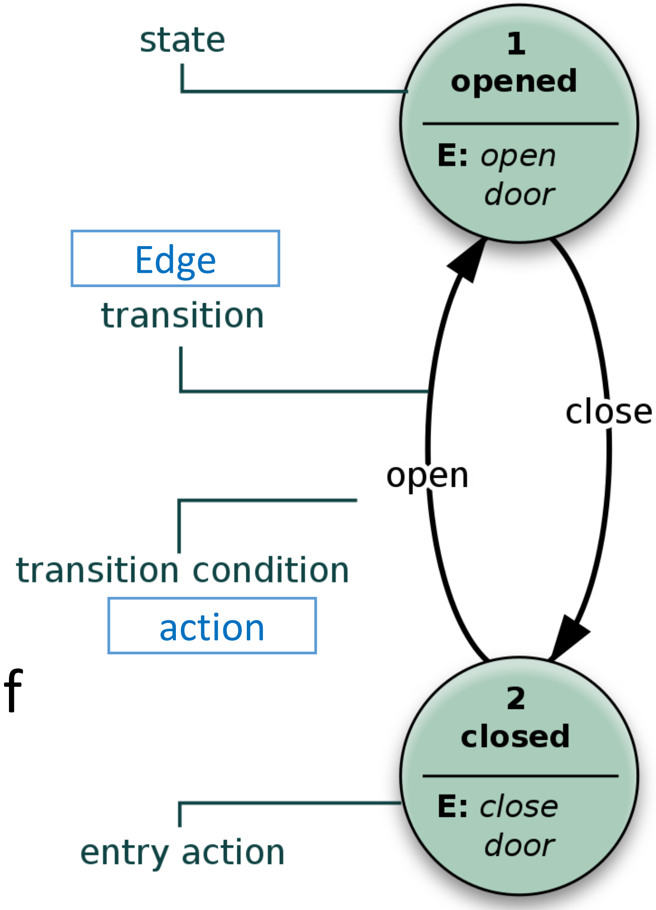


- Each operator is represented by facts that describe the state of the world before and changes to the world after an action is performed.
  - a list of **preconditions**
  - a list of new **facts to be added** (add-effects)
  - a list of **facts to be deleted** (delete-effects)
  - optionally, a set of (simple) variable **constraints**

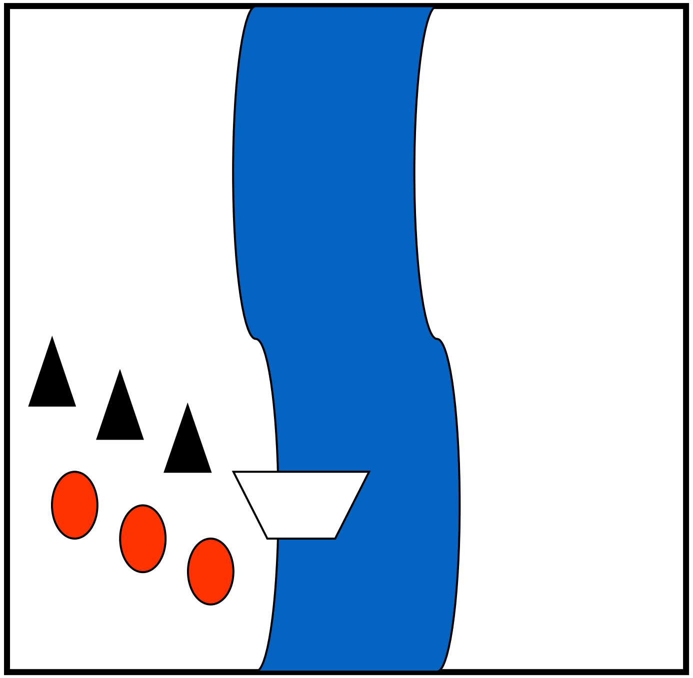
We will study this example in the Lab

State-Transition Systems Graphs

- A state-transition system $\Sigma = (S,A,\gamma)$ can be represented by a directed labelled graph $G = (N_G,E_G)$ where:
  - the nodes correspond to the states in S, i.e., $N_G=S$; and
  - there is an arc from $s \in N_G$ to $s' \in N_G$, i.e., $s \rightarrow s' \in E_G$, if and only if $s' \in \gamma(s,a)$.

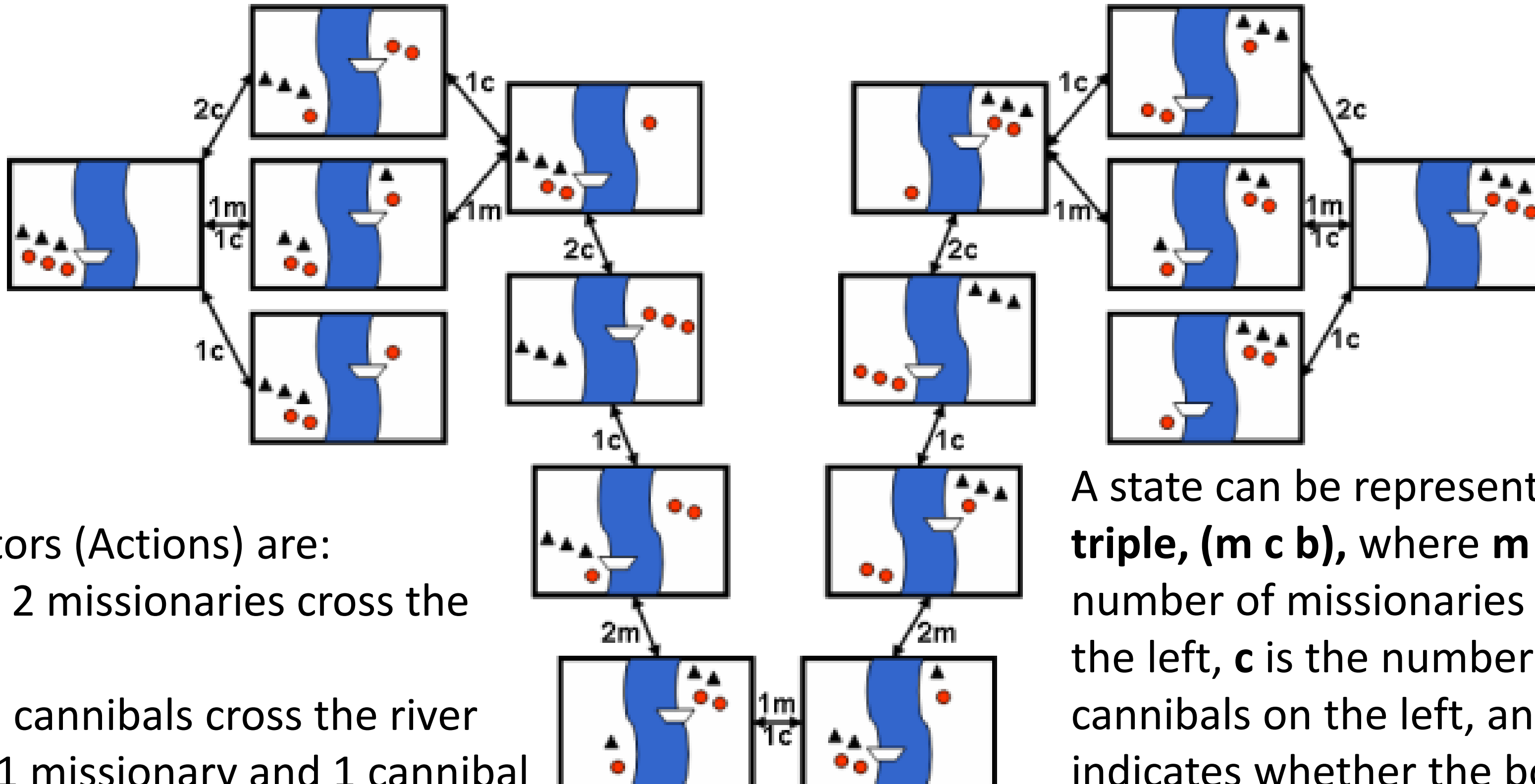


Toy Problem:
Missionaries and Cannibals



- On one bank of a river are three **missionaries** (**black triangles**) and three **cannibals** (**red circles**). There is **one boat** available that can hold up to **two people** and that they would like to use to cross the river.
- If cannibals ever outnumber the missionaries on either of the river's banks, the missionaries will get eaten.
- How can the boat be used to safely carry all the missionaries and cannibals across the river?

Toy Problem: Missionaries and Cannibals

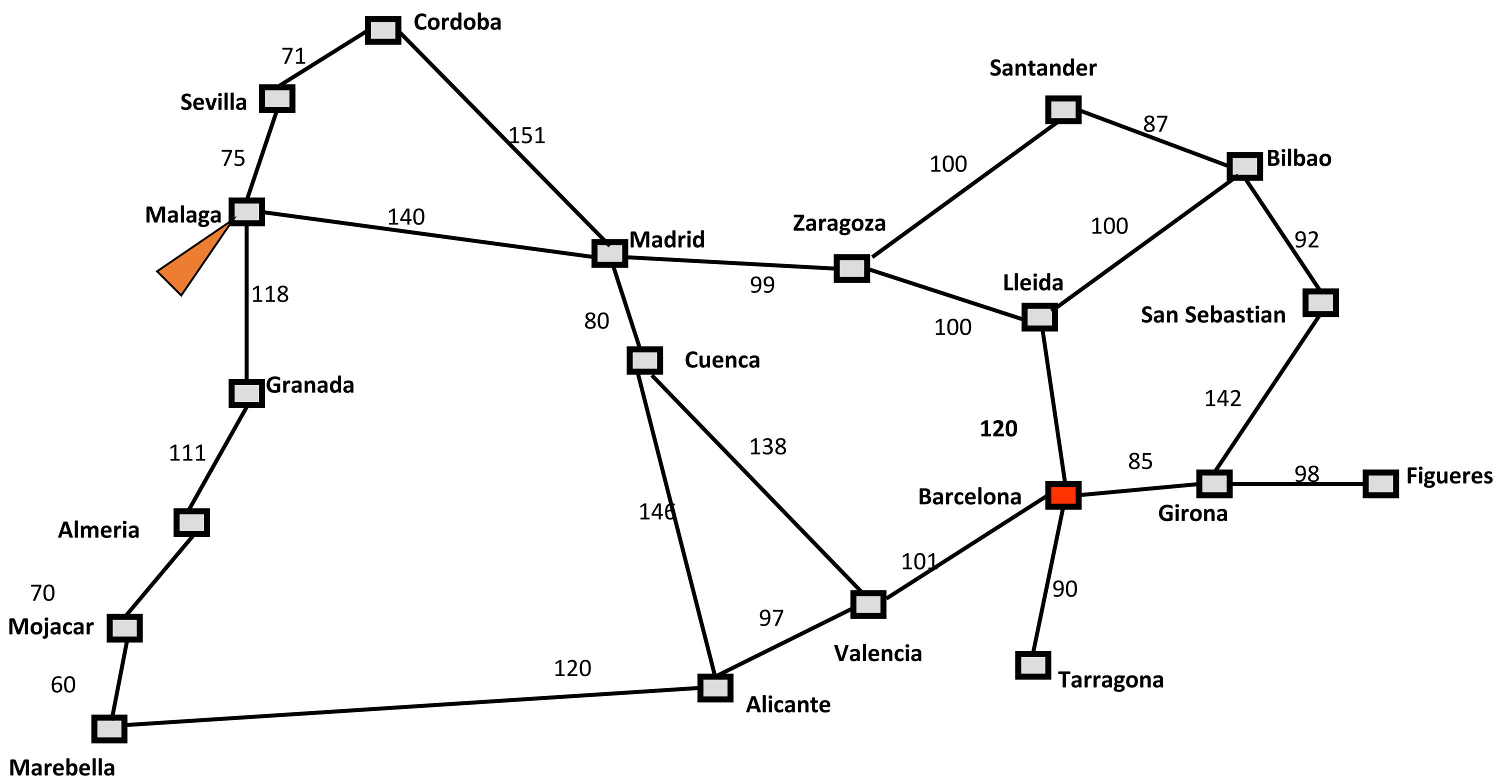


Operators (Actions) are:

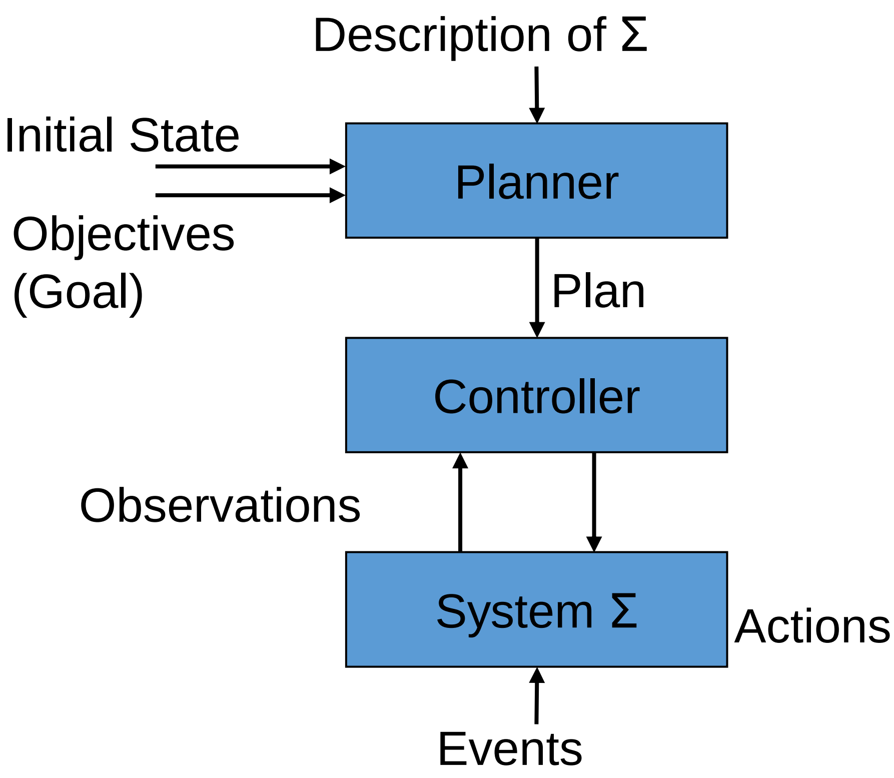
- **MM**: 2 missionaries cross the river
- **CC**: 2 cannibals cross the river
- **MC**: 1 missionary and 1 cannibal cross the river
- **M**: 1 missionary crosses the river
- **C**: 1 cannibal crosses the river

A state can be represented by a **triple, (m c b),** where **m** is the number of missionaries on the left, **c** is the number of cannibals on the left, and **b** indicates whether the boat is on **the left bank or right bank**. **The initial state is (3 3 L) and the goal state is (0 0 R)**

Real-World Problem: Touring in Spain

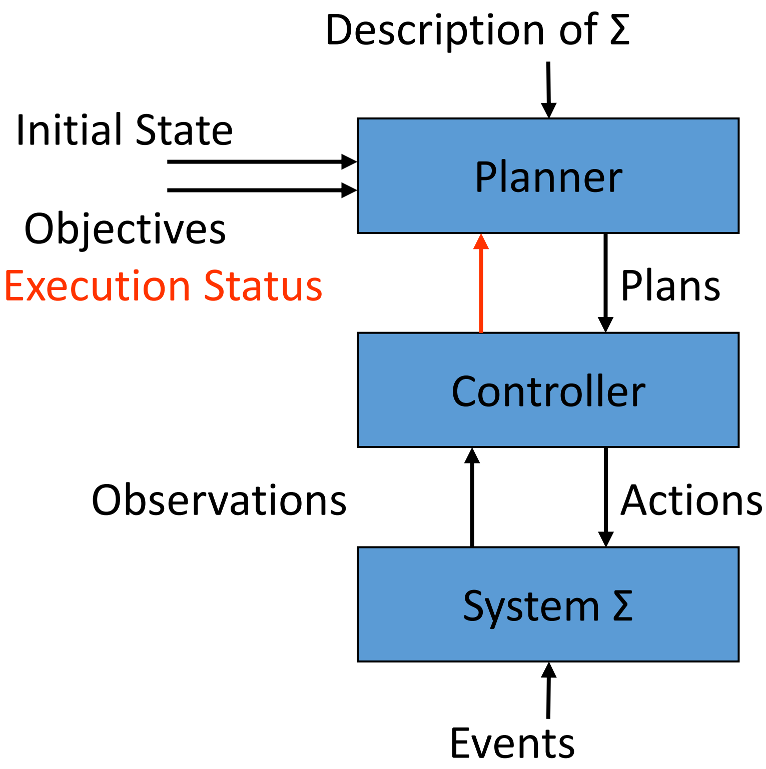


Planning and Plan Execution
Planning vs. Control



- planner:
  - given: description of Σ, initial state, objective
  - generate: plan that achieves objective
- controller:
  - given: plan, current state (observation function: η:$S\longrightarrow O$)
  - generate: action
- state-transition system:
  - evolves as actions are executed and events occur

Dynamic Planning



- **problem**: real world differs from model described by Σ
- **more realistic model**: interleaved planning and execution
  - plan supervision
  - plan revision
  - re-planning
- **dynamic planning**: closed loop between planner and controller
  - execution status

Overview

- What is Planning (in AI)?
- A Conceptual Model for Planning
- **Planning and Search**
- Example Problems

Planning as a Search Problem Problem Formulation

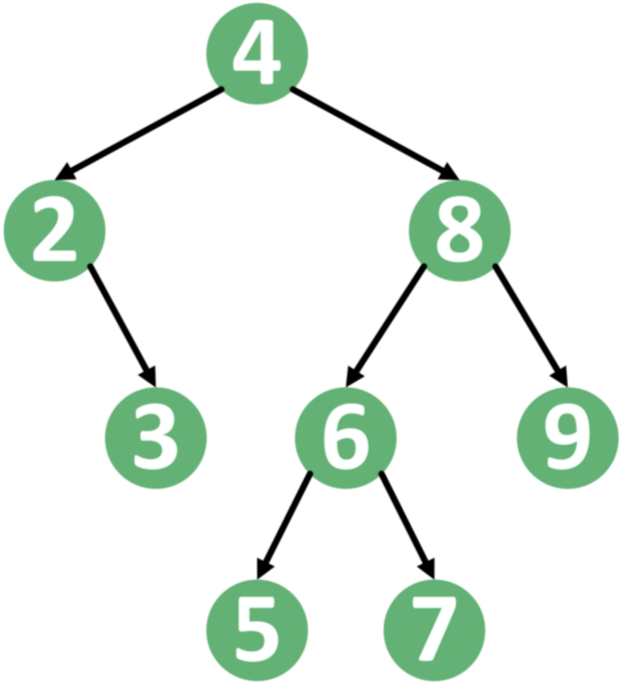
- problem formulation
  - process of deciding what actions and states to consider
  - granularity/abstraction level

- assumptions about the environment:
  - finite and discrete
  - fully observable
  - deterministic
  - static

- other assumptions:
  - restricted goals
  - sequential plans
  - implicit time
  - offline planning

Search Nodes

- Search nodes: the nodes in the search tree
- data structure:
  - *state*: a state in the state space
  - *parent node*: the immediate predecessor in the search tree
  - *action*: the action that, performed in the parent node's state, leads to this node's state
  - *path cost*: the total cost of the path leading to this node
  - *depth*: the depth of this node in the search tree



General Tree Search Algorithm

```
function treeSearch(problem, strategy)
  fringe ← { new searchNode(problem.initialState) }
  loop
    if empty(fringe) then return failure
    node ← selectFrom(fringe, strategy)
    if problem.goalTest(node.state) then
      return pathTo(node)
    fringe ← fringe + expand(problem, node)
```

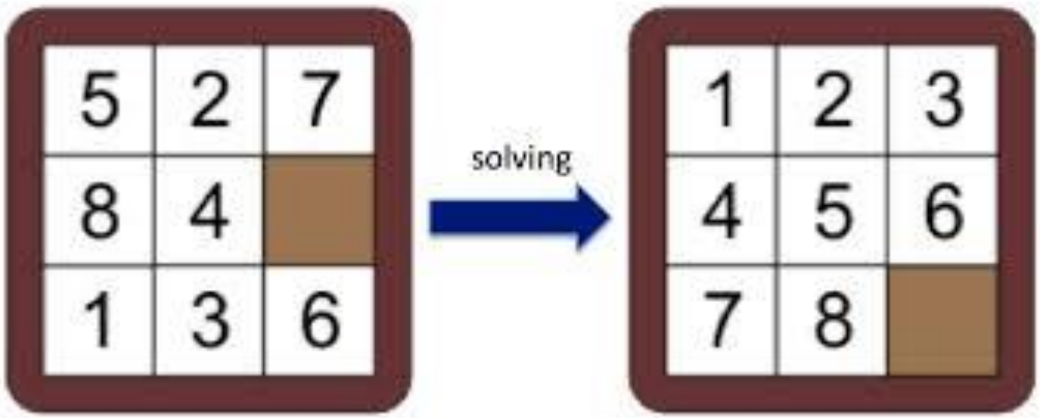
Search (Control) Strategy

- search or control strategy: an effective method for scheduling the application of the successor function to expand nodes
  - selects the next node to be expanded from the fringe
  - determines the order in which nodes are expanded
  - aim: produce a goal state as quickly as possible
- examples:
  - LIFO/FIFO-queue for fringe nodes
  - alphabetical ordering

Overview

- What is Planning (in AI)?
- A Conceptual Model for Planning
- Planning and Search
- **Example Problems**

Toy Problem: Sliding-Block Puzzle

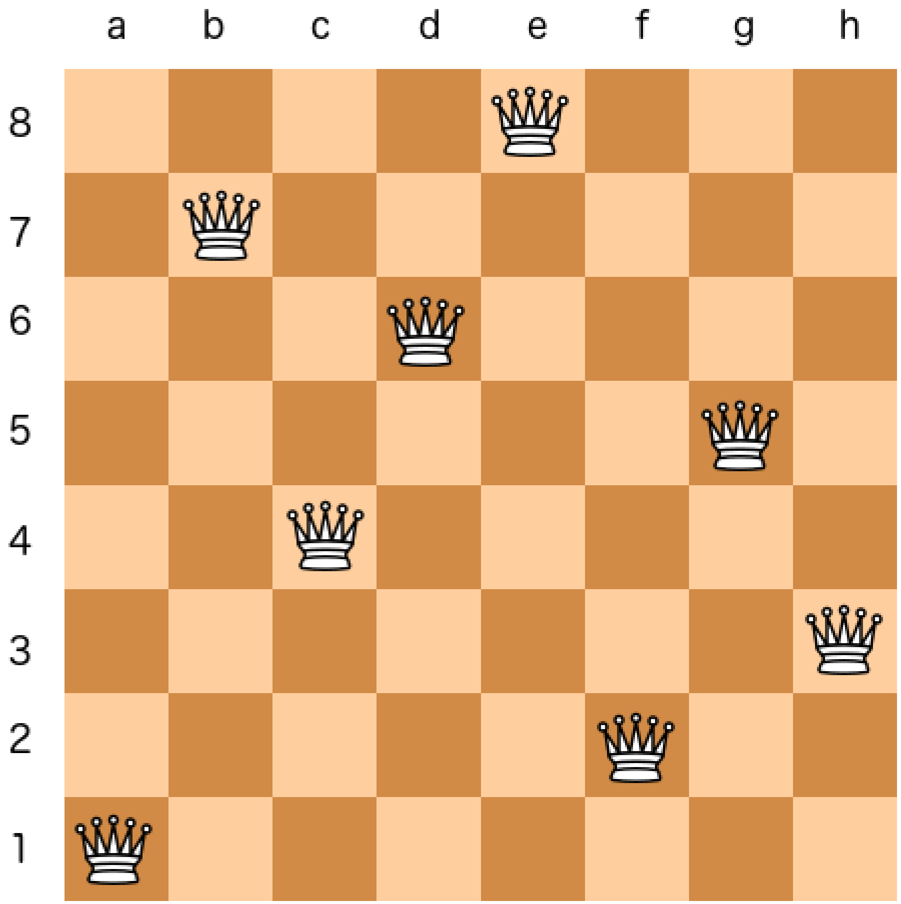


11	15	1	4
3		2	6
10	13	9	7
8	5	14	12

→

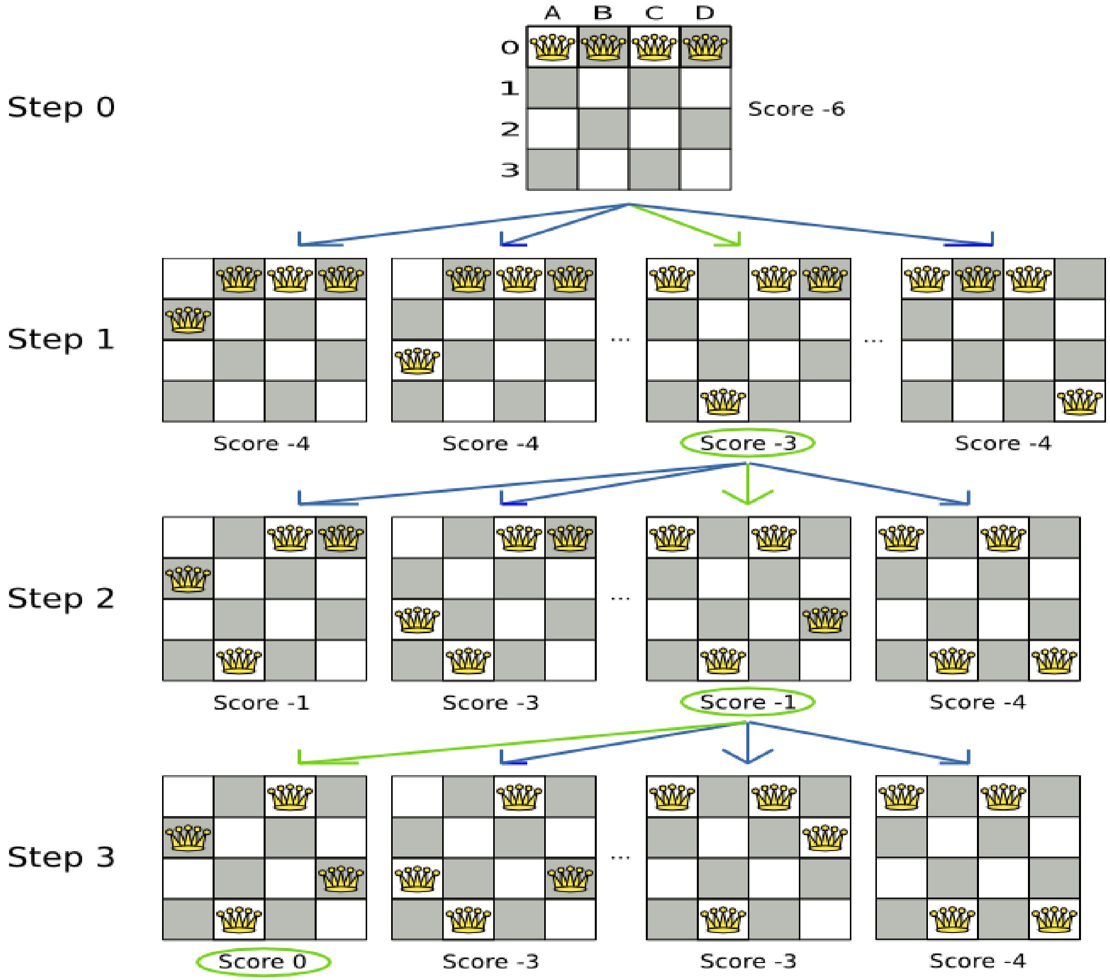
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	

Toy Problem:
N-Queens Problem



Place *n* queens on an *n* by *n* chess board such that none of the queens attacks any of the others.

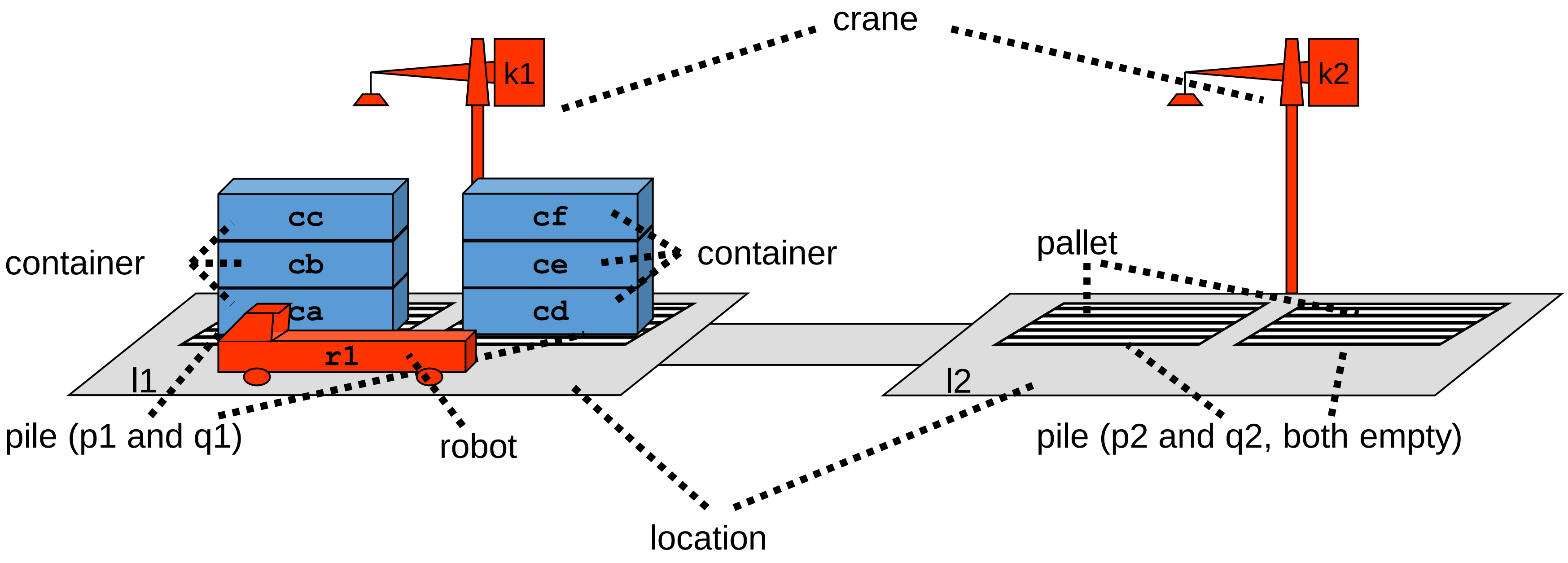
Toy Problem:
N-Queens Problem



Dock-Worker Robots (DWR) Domain

- aim: have one example to illustrate planning procedures and techniques
- informal description:
  - harbour with several locations (docks), docked ships, storage areas for containers, and parking areas for trucks and trains
  - cranes to load and unload ships etc., and robot carts to move containers around

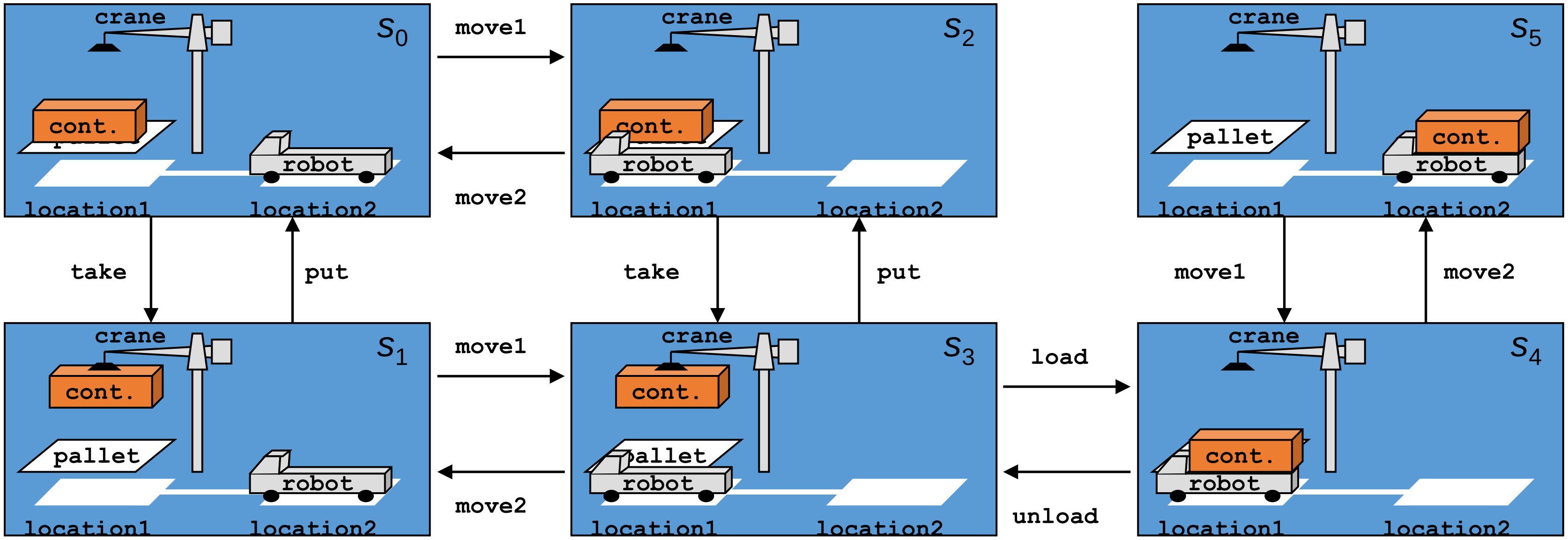
DWR Example State



Actions in the DWR Domain

- `move` robot *r* from location *l* to some adjacent and unoccupied location *l'*
- `take` container *c* with empty crane *k* from the top of pile *p*, all located at the same location *l*
- `put` down container *c* held by crane *k* on top of pile *p*, all located at location *l*
- `load` container *c* held by crane *k* onto unloaded robot *r*, all located at location *l*
- `unload` container *c* with empty crane *k* from loaded robot *r*, all located at location *l*

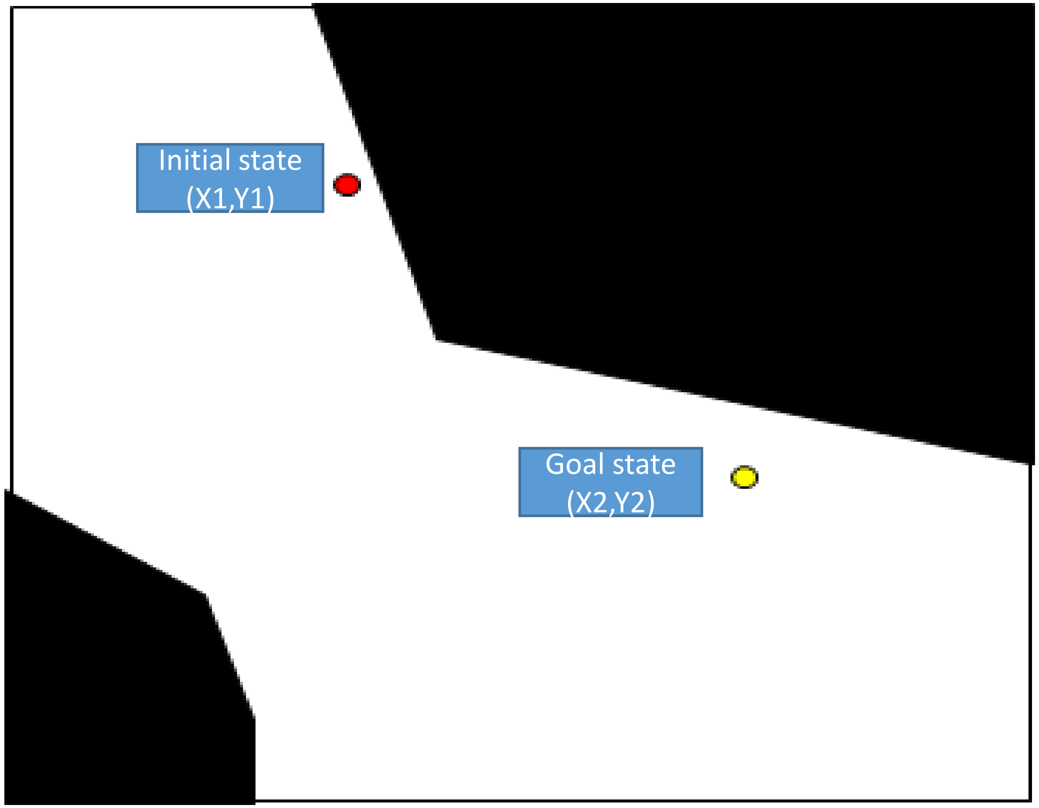
State-Transition Systems: Graph Example



Planning Examples

❑ 2D path planning for omnidirectional robot

- What is model M^a ?
- What is Belief b_c^a?
- What is *Belief* b_c^w?
- What is the cost function ***C*** ? (Path)
- What is the goal ***G***?



Planning Examples

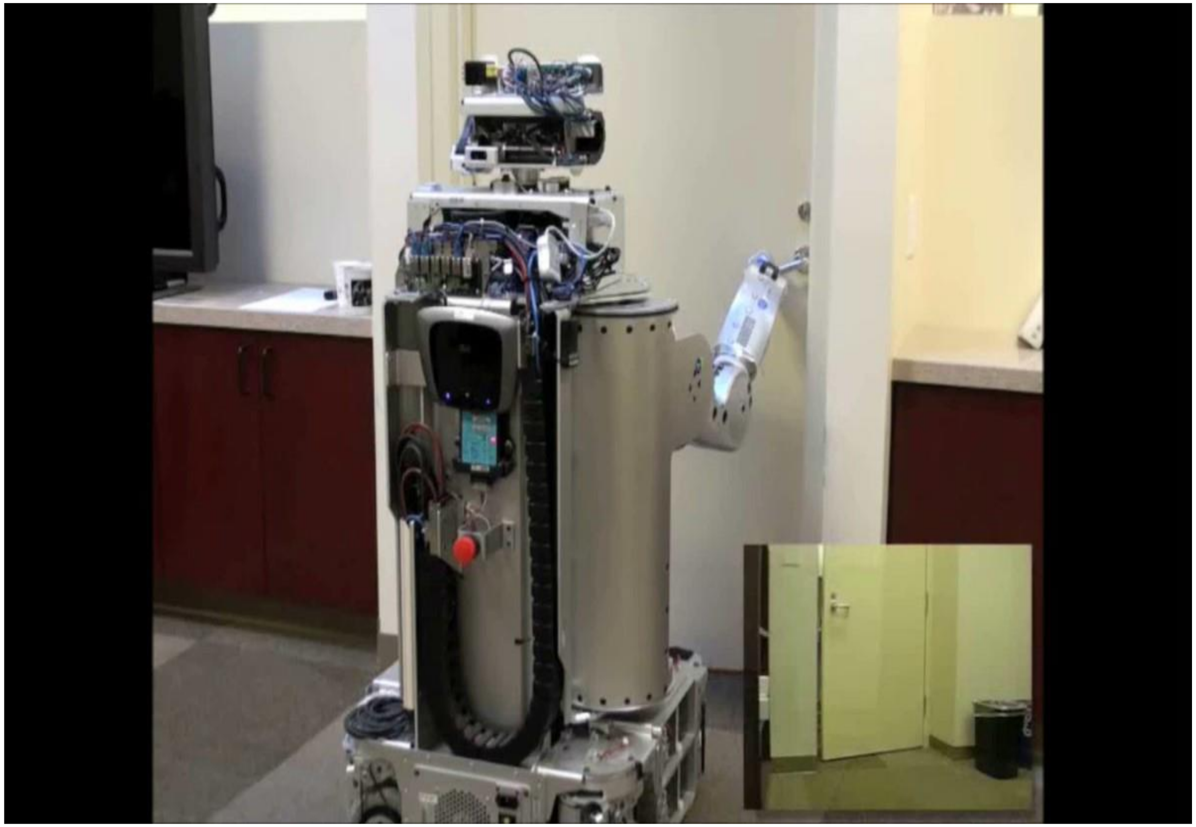
❑ **5D (x,y,z,direction,time) path planning for autonomous drone among people :**

- What is M^a ?
- What is b_c^a?
- What is b_c^w?
- What is C ?
- What is G?

Planning Examples

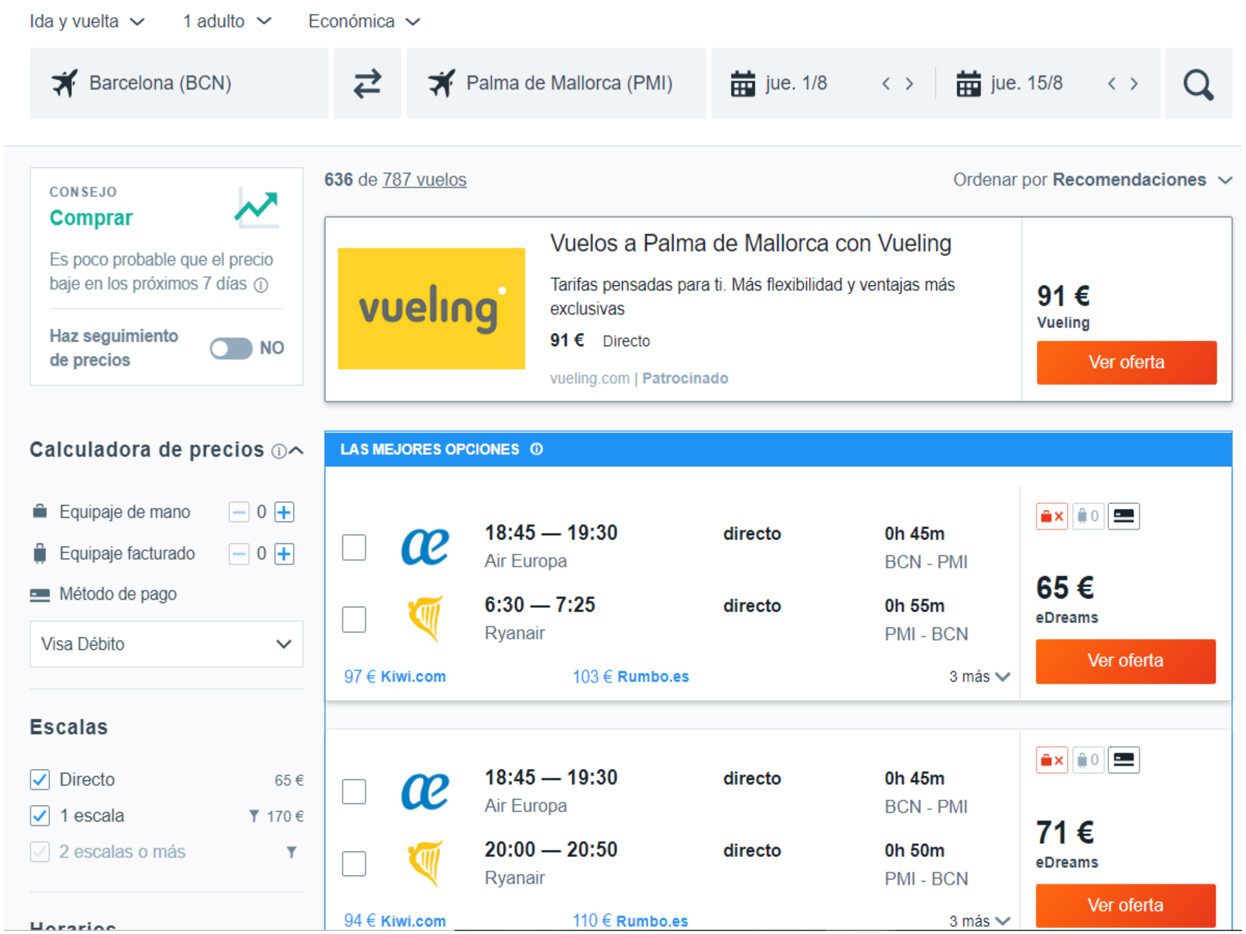
- **Motion planning for a mobile manipulator PR2 opening a door**

  - What is M^a ?
  - What is b_c^a?
  - What is b_c^w?
  - What is C ?
  - What is G?



Planning Examples

- Planning a travel from Barcelona to Palma Mallorca

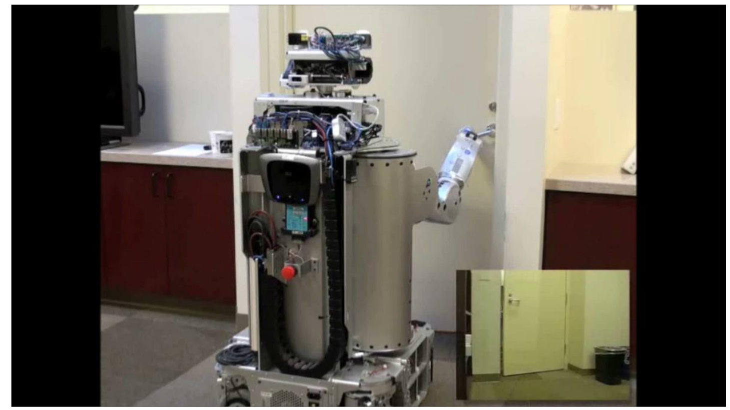


- What is M^a ?
- What is $b_c^{\,a}$?
- What is $b_c^{\,w}$?
- What is C ?
- What is G?

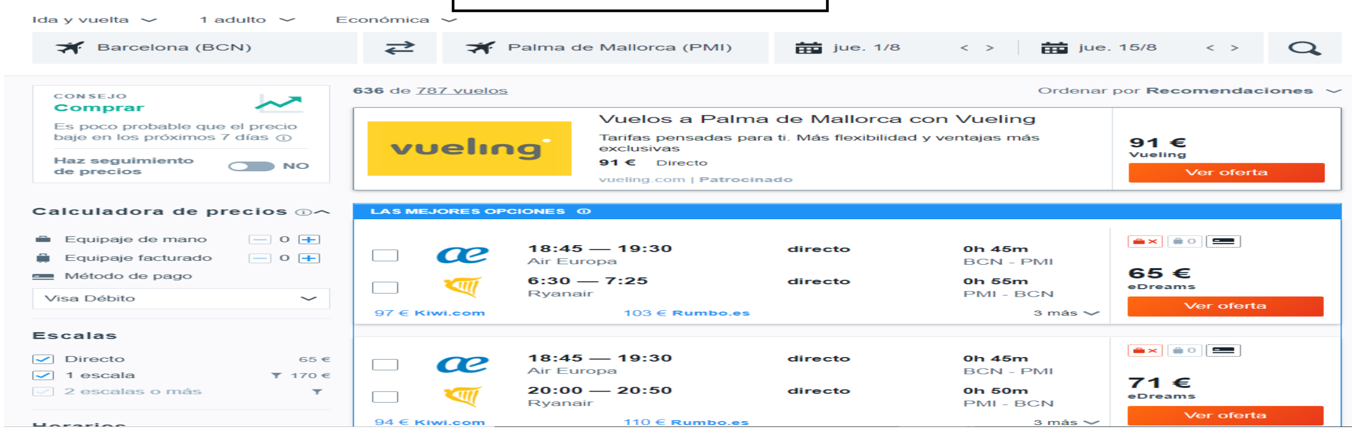
Continuous vs. Discrete vs. Hybrid Model

Continuous

Hybrid



Discrete



End