AI Course Fall 2015:
Exercise 1: PEAS, Basic Search and Informed Search

Task 1 PEAS Problem and Environment Categorization

a) For each of the following problems, give a PEAS description of the task environment and characterize it in terms of the environment properties given in the lecture

- writing an exam
- practicing tennis against the wall
- bidding on an item at an auction (e.g. for an old carpet at Sotheby's)

Writing an exam

Performance Measure	Environment	Actuators	Sensors
#points/grade or boolean pass or not	Exam room, paper for writing on, Examinator	Hand with pen, speech production for asking	Eyes, ears, tactile sensors at the hand would be too fine

Properties of the Environment:

- fully observable (if the tasks are well formulated and there is no need to speculate what the examiner wanted to know.
- Deterministic, assuming that the grading happens without random element and the grading is reproducible
- Episodic, if we ignore that one could get used to writing exam and we also ignore the knowledge gain from courses
- static, if there is no time pressure finishing the exam
- discrete, if we assume that there are distinct possible answers to questions.
- single agent / two agents: depending on the active or passive involvement of the examiner (and nobody can cheat by looking into others' solutions)

Play tennis against the wall

Performance Measure	Environment	Actuators	Sensors
Maximize time during which the ball is in the air Number of bumps against the wall	In door environment, wall	racket	Eyes, ears, tactile sensors at the hand with the racket

Properties of the Environment:

Properties of the Environment:

- fully observable (at any time we can observe the ball)
- Deterministic with a perfect wall and sufficiently fine adjustment of racket; stochastic if the wall is not perfect
- Episodic
- static (without wind)
- continuous
- single agent

Bidding in an auction at Sotherby's
we consider the complete auction, not just one bidding action

Performance Measure	Environment	Actuators	Sensors
Price for which I get the item, whether I get the item	Other bidders, auctionator	Message/signaling device for bids	Sensor that gets information on the current highest bid – eye/ear…

Properties of the Environment:

- partially observable, as we cannot see into the heads of the other bidders, so that we knew when they stop increasing the price.
- Deterministic (if my bid is seen and reacted to – which is actually what the auctioneer wants, so he/she will take care that no bid is lost). If I hand in the highest bid, I get the item
- sequential (it is not about handing in bids); after getting the carpet, I have less money for the next auction (and my living room is full of carpets)
- dynamic: Others may bid while I think about bidding
- discrete (if you are allowed to increase bid with certain amount)
- multi-agent

➔ **Properties of a Task Environment: there is not true environment characterization for each of the tasks without looking at the actuators and sensors of the intelligent entity.**

b) Consider the PEAS environmental properties: In which environments you can use search as a problem solving method?

We have seen search for discrete, fully observable, static, single and multiagent environments. There are variants for stochastic environments – we basically integrate nodes with probabilistic transitions. There are also algorithms that search in "Belief" space for handling not fully accessible situations. A state is then a set of possible states. As we know what happens (deterministic) when doing an action, we can determine the next belief state from the current one. Whether it is episodic or sequential is not important for applying search; Dynamics: We assume that the start (or goal) state is not changing while we are searching for a sequence of actions to reach a goal state – for that we would need to add an additional control layer telling us whether the assumed (start) state still holds

The search that we use clearly is based on **discrete states**

Task 2 Problem Modeling

In the Towers of Hanoi Problem the task is to move the complete stack from one rod to another rod, with only one disk can be moved at one time, only the top disk of a stack can be moved and disks can only be placed on larger disks:

This is the start situation of a 6-disk Towers of Hanoi problem.

Model the problem as a search problem and illustrate how you would use basic uninformed search for it (don't search for more than 2-3 steps)

Can you suggest an admissible heuristic for applying A* for solving this problem.

Modeling the problem as a Search Problem:

- **Initial State:** the situation represented graphically above or something like:
  <(lila,red.orange,yellow,green,red),(),()>

- **Actions**: the action are a list of operations that are possible in a state. Here we express the conditions for actions. In general we can generate all possible actions for the rods and then filter: In the start situation we have **actions(S0)={Rod1->Rod2 ,Rod1->Rod3 }** For each state we have at maximum than 6 possible actions -

- **Transition model gives for each state and actions the resulting state. In our simple world, the resulting state is always unique. In in-deterministic worlds, that would be different…:**
  RESULT(s0,Rod1->Rod2)= <(red.orange,yellow,green,red),(lila),()>
  RESULT(s0,Rod1->Rod3)= <(red.orange,yellow,green,red),(),(lila)>
  These two are the successor states that we have to consider then.

-
- **Goal Test: are all disks on Rod3?**
- **Path Cost Function**: let's assume simply uniform costs

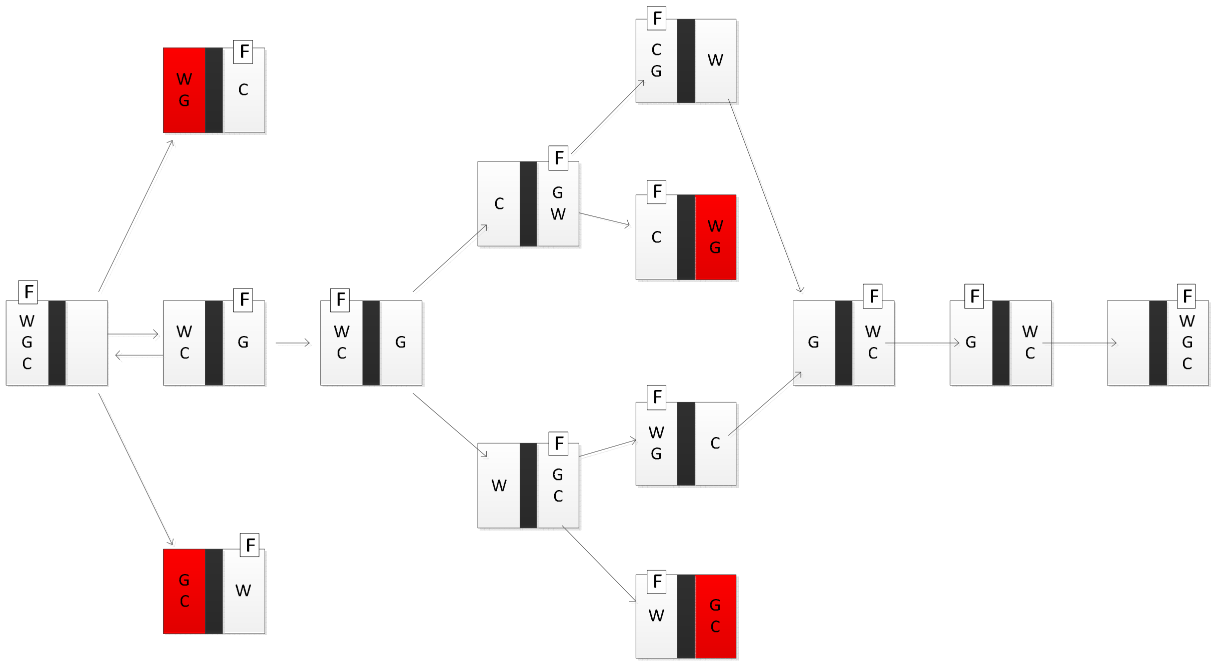
With that formulation, we simply can use our algorithms: using the initial state as start, the goaltest for checking whether we are finished. The sucessors are determined based on what actions are possible in which state and to what next state the actions then lead. We do not need to change anything at the algorithms.

Heuristics? Admissible A* Heuristic – costs must be always underestimated. The heuristic must allow us to compare states and indicate which next state is more promising than the others?
Checking how many disks are already on the 3rd rod, might not be the most accurate, but it is clearly an admissible heuristic.

Task 3 Task Search Space and Search

A classical problem for illustrating search is the Farmer's Problem: A farmer has the problem to transport a goat, a wolf and a cabbage to the other side of a river. His boat is small, so it can only carry him and one other entity - either goat, wolf or cabbage. The problem is, that wolves feed on goats and goats love to eat cabbage which immediately happens if the farmer is not present. That means, if he leave for example the goat and the cabbage on one side of the river, transporting only the wolf, the cabbage will be gone when he returns.

Draw the state space graph and give a solution to the problem.



F=Farmer
W=Wolf
G= Goat
C= Cabbage

Task 4 A* Heuristics

In the lecture, we shortly discussed (will discuss on 19th) two heuristics for the 8-Puzzle. In the 8-Puzzle 8 tiles with numbers on them have to be brought into a particular configuration on a grid which allows just to move one tile onto a free space.

The following pictures for the start and goal state:

4	3	1
6	5	2
7		8

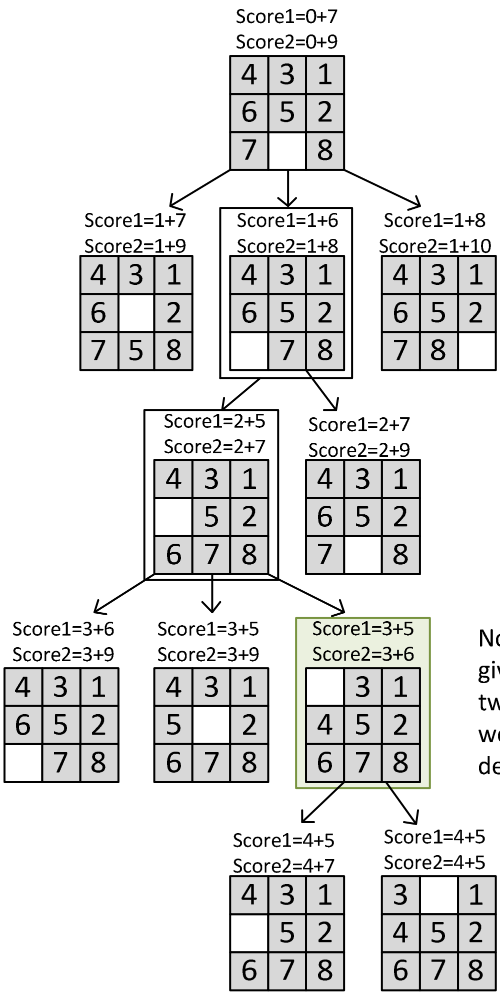
Start state

	1	2
3	4	5
6	7	8

Goal state

Two heuristics are possible:
$h_1(n)$ = number of misplaced tiles
$h_2(n)$ = sum of Manhattan distance for each misplaced tile to its destination place

Perform an A* search until depth 4 (inclusive) for each of the heuristics. What can you observe? What heuristic would you use?



Now we are at a point that h2 still gives enough guidance, but with h1 two states are equal, so practically we would need to do a random decision which next.

From that state on both heuristics will return the same values.