Departamento de Informática
Universidad Carlos III de Madrid

Grado en Ingeniería Informática

Artificial Intelligence

May 2017

General instructions

- Time assigned to the exam is **2.5 hours**
- Teachers will not answer any question about the exam
- You cannot leave the classroom during the exam, unless you have finished it
- Exams cannot be answered using a pencil

General questions

We expect very short answers.

1. (3 points) A company sells a cleaning system based on two robots. Given a map of a building floor of $M \times N$ tiles and the position (tiles) of fixed objects on that floor (tables, chairs, machines, ...), the system has to compute a two-robots plan. The plan decides at each time step which movement will perform each robot. Actions are deterministic. Available movements are to horizontal or vertically adjacent tiles: north, east, south and west. Consider that the two robots can either move (each one might use a different movement) or stay in the same position at each time step. Robots cannot move to an occupied tile by an object. Also, the two robots cannot move to the same tile or any robot move to the position of the other robot if the other robot decided to stay in its tile in the next time step. When a robot moves into a tile, that tile is cleaned. The objective is to clean all tiles that do not have objects.

   - Describe a possible structure of the working memory. Use any representation formalism.
   - In case the current state is the one represented in Figure 1, formalize the contents of the working memory according to the previous item.

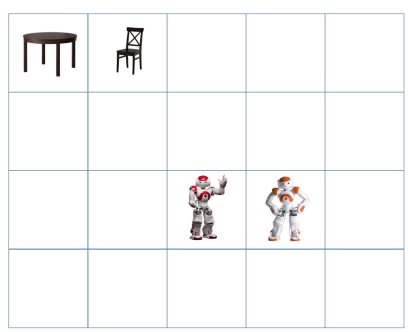

Figure 1: Example of a two robots scenario.

   - How many different states can we create with that working memory?
   - How many different valid rules can we create?
   - How big would the conflict set be given a particular working memory in the worst case?

2. (3 points) In the previous question, and given the same constraints, we would like to compute an optimal solution that minimizes the number of time steps in the solution.
   - Mention two different algorithms to solve this task. At least one of them has to perform heuristic search. Enumerate their theoretical properties in relation to this problem.
   - Define their parameters in case they have.
   - Show the expansion (not the generation) of the first three nodes using any of the algorithms that performs heuristic search in the initial state described in Figure 1.
3. (4 points) Let us assume that there is some stochasticity in the robots movements. The probability of a single robot to move to an adjacent tile is 0.8, while the probability of remaining in the current tile is 0.2.
   - Given these assumptions, which kind of algorithm would help model the joint movement of the two robots in order to clean all tiles without objects taking into account the stochastic component?
   - Formally describe each of its components.
   - What components would change (and how) if
     - the probability of moving to the next tile changes to 0.7?
     - there are three robots?

Question about the project (1 point)

Explain in detail the implemented learning process of Pacman. Also, explain why there are sometimes situations when the behavior is random.

Solutions

Theory questions

This is just one possible valid way of answering the exam questions. It does not mean they are the only valid answers.

First question

- A possible representation is an $M \times N$ array with the contents of each position in the set C={object, robot1, robot2, dirty, clean}. Another representation using predicate logic would be a predicate `position(X,Y,Contents)`, where $0 \leq X \leq M-1, 0 \leq Y \leq N-1$ and Contents any value in C.

- In the first representation, the array contents should have the corresponding value at each position. In the second one, there would be a ground predicate for position. You are supposed to be more specific and provide some examples.

- Given that each position can have a value in C, the size of the state space would be: $5^{M\times N}$. Many of those states would be invalid states (those with less or more than two robots).

  A more precise figure for the size of the state space would consider that the first robot can be in any of $M \times N - O$ positions where O is the number of positions occupied by objects (we assume a fixed configuration of objects). For each of these positions, the second robot can be in any of the $M \times N - O - 1$ remaining positions. Then, the remaining positions (not occupied by robots or objects) can be in any of two states, clean or dirty. Therefore, there will be $2^{M\times N-O-2}$ different configurations for them. Therefore, the total number of valid states would be:

$$(M \times N - O) \times (M \times N - O - 1) \times 2^{M\times N-O-2}$$

- Since at each time step, each robot can either move north, east, south, west or stay, and each rule should contemplate the movement of the two robots, we would have $5^2 = 25$ different rules. This figure assumes generic rules, as, for instance, move robot1 north and robot2 south, independently of the actual position of both robots. If you consider all instantiated rules, you would have to account for the number of different actions where both robots can move (or stay) and where. You could be more specific and provide some examples.

- In the worst case, the production system could execute all rules with one instantiation per rule, so $5^2 = 25$.

Second question

- Since we are required to compute optimal solutions, we should use an admissible algorithm as breadth-first search (costs are assumed to be unitary), Dijkstra (equal to breadth-first search in the case of unitary costs), or A*. All are complete and admissible algorithms.

- The only algorithm with parameters is A* that requires to define the cost function and the heuristic function. The cost function, $g(n)$, will be defined in terms of the cost of performing the robots movements. Thus, for each operator that moves the two robots, the cost of the operator would be 2. For each operator that moves only one robot, the cost of the operator would be 1. And the operator where the two robots stay in the same position its cost would be 0. $g(n)$ is defined as the sum of all operators from the root node to node n.

  In relation to the heuristic function, $h(n)$, at each time step, there are D dirty positions (not cleaned already). The minimum time steps to clean those positions would require to move at least one robot to the closest position to any of them and then one time step for each couple of remaining dirty positions ($D-2$ given that while a robot is cleaning the closest dirty tile, the other robot could also be cleaning another closest dirty tile). So,

$$h = \lfloor \frac{D-2}{2} \rfloor + \min_{d\in D, r\in\{\text{robot1, robot2}\}} \text{distance}(d, r)$$

  where distance is computed as the Manhattan distance between the position of tile d and robot r.

- You are supposed to provide an example of a search tree.

Third question

- Stochastic problem that considers actions: MDP.
- MDP: $\langle S, A, T, R \rangle$

  where

  - S: set of states equal to the same set of states in the first question
  - A: set of actions equal to the same set of states in the first question
  - T: transition function, $T(s, a, s') = P(s' \mid s, a)$

    you are supposed to provide some examples as if s is the one in Figure 1, action is (north, north), and s' is the one where the robots are in (2,2) and (3,2) (assuming (0,0) to be in the bottom left corner), then

    $$P(s' \mid s, a) = 0.8 \times 0.8 = 0.64$$

    And if s' is the state where the robots are in (2,1) and (3,2), then

    $$P(s' \mid s, a) = 0.2 \times 0.8 = 0.16$$

  - R: reward function (or cost function as in class), $R(s, a, s')$. We can define a cost function of 2 per action where the two robots move, 1 where only one moves and 0 otherwise.

- In that case

  - if the probability of moving to the next tile changes to 0.7, then the only component that changes is $T()$. You would have to specify how.
  - if there are three robots, then all components change. You would have to specify how.