Search and Planning

Exercise 1 (8 points)

A repair robot is equipped with an arm handling a screwdriver and another arm to manipulate objects. The two arms can not operate simultaneously. The robot must repair a refrigerator, which is already placed in the working space of the robot, by replacing the broken compressor. The refrigerator is still on and a new compressor is available. The compressor is accessible by removing the backplane, which is held by 2 screws. Before extracting the compressor, the fridge must be powered off. The two actions to unmount the broken compressor and to mount the new one into the fridge can be modelled as primitive actions without further decomposing them. The goal is to have the fridge repaired, fully reassembled and powered off.

(a) Describe the domain in PDDL;

(b) Describe the problem in PDDL;

(c) Discuss the forward planning process to reach the goal, using a *perfect* heuristic that gives for each state the number of steps to reach the goal; for each step, show the current state, the applicable actions and the state resulting from the application of the chosen action.

Exercise 2 (4 points)

Describe the language for HTN planning. In particular, explain the notion of *reachable set*.

Exercise 3 (4 points)

Describe the algorithm A*, focussing in particular on correctness, completeness and memory requirements. In addition, address the limitations of A* and possible approaches to mitigate them.

Knowledge Representation

Exercise 1 (8 points)

Given the following knowledge:

1. *Gourmand students eat at breakfast bread with nutella.*
2. *Regular students eat at breakfast bread only(*).*
3. *Students that are not gourmand are regular.*
4. *Francesco is a student.*

(a) Define a vocabulary (i.e., constant, function and predicate symbols) and represent the following sentences in first order logic.

(b) translate the sentences in clausal form

(c) derive that Francesco eats at breakfast using resolution.

(*) In the representation of the sentence consider also the constraint only, even though it is not needed in the derivation.

Exercise 1d

Write one model for the above knowledge base. Are there other models?

Exercise 2 (4 points)

Describe all the known methods for verifying *logical consequence* in propositional logic.

Exercise 3 (4 points)

Write a PROLOG program that, given two lists of integers `INDXS` and `VALUES`, returns a list whose first element is the value stored in the position corresponding to the first element of `INDXS` of the list `VALUES`. For example, given `INDXS=[2,1,4,3]` and `VALUES=[2,4,6,8]`, the output is `[4,2,8,6]`. You may start by defining a predicate that, given an integer `I` and a list `L` returns the value of the element of the list `L` in the position `I`. Explain what happens if the position `I` does not correspond to an element of `L`.