89-674: Introduction to Intelligent, Knowledge-Based, and Cognitive Systems

Spring 2018: EX2

Due at 23:59:59, Tuesday, April 12th, 2018.

Assignment

This assignment is divided to 2 parts:

Part 1: Write a domain and a problem definition in PDDL

In this part you need to write a domain definition and a problem definition for the following problem:

There is an attacker agent that can take over any computer currently connected to a computer that is already controlled by it. It has 4 actions of attacking, for 4 kinds of operating systems: windows7, windows10, linux mint, ubuntu16. There are 10 computers: 3 have windows10, 3 have windows7, 3 have ubuntu16 and 1 has linux mint. Not every machine is necessarily connected to all others.

You need to write the domain description and following problem definitions:

1. The agent is currently in control of a windows10 computer and wants to take over the computer with linux mint. The network connectivity is such that it needs to go through all the computers to reach the linux mint computer.

2. The agent is currently in control of a windows10 computer, and wants to take over at least one computer, of each operating system. Machines that have the same operating system are connected as a clique, but only one machine from each such clique is connected to another clique. No one machine can be connected to two cliques.

Part 2: Write a smart re-planner in pddlsim

Write an agent that needs to plan only if needed, that is if it has multiple possibilities.

Your agent needs to be able to receive both the domain and the problem definitions has arguments from the command line.

Your code will be run with the following command:

python smart_replanner.py <domain_file> <problem_file>

You can test your code on the domain and problem files given in the piazza, but we will test them on other problems in the domains given.

Bonus for this part (optional): Extend the agent to also makes decisions on what action to take not necessarily based on planning but using a different mechanism. Such mechanisms can be:

- Heuristics for preferring an option (e.g., Manhattan distance, or its equivalent in fluents == number of fluents not having the goal value)
- Look-ahead – instead of planning fully ahead just check a few (k) steps ahead to decide if one is alternative seems better than another.
- Any other method you can think off.

You can make domain specific heuristics and check the domain of the problem at the initiation of your agent at each run.

If you do the bonus explain in the README file what was your method of choosing the next action.

The Rules

1. Your code will be run on an ubuntu16 make sure your code works on a this O.S. If your code works on your O.S that isn't ubuntu16 but won't work on our computer you will not be able to appeal on your grade.
2. Your work should be performed independently.

What to Hand In

You need to hand in at least 5 files.

- attack_domain.pddl - The pddl file for part 1 domain.
- attack_problem1.pddl - The pddl file for part 1 problem1.
- attack_problem2.pddl - The pddl file for part 1 problem2.

- smart_replanner.py – that runs your agent with a given domain and problem.
- README – Contains your name and your ID number and any explanation on your code you wish to give. If you do the bonus this file will also include your description on how you choose an action

Do not hand in any pddl files other than the ones you created.

Finally, zip all the necessary files and submit in https://submit.cs.biu.ac.il/cgi-bin/welcome.cgi .

Good luck!