Principles of AI Planning

Prof. Dr. B. Nebel, R. Mattmüller
N. Mayer
Winter Semester 2012/2013

University of Freiburg
Department of Computer Science

Exercise Sheet 1

Due: November 2, 2012

Exercise 1.1 (State space sizes, 5 points)

In the following we describe two planning problems. For each, determine the sizes of the respective state spaces, i.e., the number of possible different states. How much time would it take to traverse the whole state space if visiting one state took $1\mu s$ ($= 10^{-6}s = 0.000001s$)?

(a) "Boethiah's Delivery Service" serves 4 towns in Skyrim: Whiterun, Solitude, Winterhold and Windhelm. In each of these towns, errand boys travel between all districts (within their own town). Whiterun consists of 8 such districts, Solitude of 2, and Winterhold and Windhelm each have 12.

Furthermore, in each town one of the districts contains a landing place for dragons, the exception being Whiterun, which has 3 such districts.

Currently, the delivery service is entrusted with delivering 30 packages to other districts or towns. At its disposal are 3 dragons as well as (number of districts)/2 errand boys per town.

Errand boys can pick up packages if they are located in the same district. Dragons can pick up packages if both the dragon and the package are at the same dragon landing place (Dragons cannot land in any other districts[1]). Unloading packages is subject to the same restrictions.

Surprisingly, in all of Tamriel's recorded history since the very first Era there has never been a documented case of any errand boy or dragon ever being overencumbered (thus being unable to pick up any more packets). Their strength seems to know no bounds ...

As ancient beings of great power, each dragon carries a unique name, and bureaucracy forces every errand boy to choose a distinct name for himself as well. For this reason, it is indeed very important *which* dragon(s) and/or *which* errand boy(s) are at a particular location.

(b) There is a rather small house in which there are two distinct rooms named *RoomA* and *RoomB*. The only resident, an incredibly old and at least slightly quirky man, is an ardent collector of balls, no two of which are alike. All of his collection is stored in *RoomA*. Unfortunately, that room now has to be repainted immediately, so the ball collection has to move to *RoomB*. The painters are quite sure, yes Sir, that this is not their responsibility, and since the old man is too weak to do it himself, he acquires a robotic companion to move the balls. The robot, being a dapper little fellow, albeit somewhat undersupplied in the creativity department, has two arms called *leftArm* and *rightArm*. Which each of these, the robot can pick up a ball (and drop it again). Finally, the robot is also able to move between the two rooms.

The arms are named and thus clearly distinguishable from each other, i.e., it matters in *which* arm a ball is being held.

[1] Unless for demolition work, but Boethiah leaves that field to others.

Exercise 1.2 (PDDL and Pyperplan, 5 points)

Your quirky neighbor left you his entire fortune in his will. You don't quite know what you expected, but it consists of a rather small house with two rooms. Curiously, the rooms have names, *RoomA* and *RoomB*. You are surprised, but a house is a house, so you decide to throw a party in *RoomB* to celebrate you new status of homeowner.
Unfortunately, this room seems to be stuffed with a collection of 6 balls (your neighbor was really quite quirky, you decide). Luckily, *RoomA* is spacious enough to move the balls there. As you plan your next steps, you stumble upon a robot which seems to be exquisitely fit for moving balls between two rooms (you wonder if there is a stronger word for "quirky"). As you try to activate the robot, you discover a strange problem: There is a perfectly fine third arm that is not to be found anywhere in the planning system – the somewhat amateurish PDDL specifications neither consider the third arm nor the mint condition *seventh ball*.
It's already friday and the party is planned for tomorrow. The manufacturer does not respond, so you roll up your sleeves and set about reconfiguring the PDDL specs yourself.

(a) Download the files for the *gripper* planning domain from the lecture webpage's exercises subpage. You can find the domain description in the file `gripper-domain.pddl` and the problem itself in the file `gripper-problem-two-arms-six-balls.pddl`. Solve the problem using Pyperplan (see webpage) and fill out the following table of plan lengths and node expansions for the given combinations of search algorithms and heuristics.

	blind	**hmax**	**hadd**	**hff**
Greedy Best First				
A-Star				
Breadth First Search				

(b) Add a third arm (gripper) called `middle` to the problem description, as well as one more ball `ball7`. `ball7` is initially located in `roomb` and is to be moved to `rooma`, just like the others. Solve the modified problem using Greedy Best First Search with the FF heuristic and send the new problem description file and the solution file generated by Pyperplan to nikolaus.mayer@merkur.uni-freiburg.de.

You can and should solve the exercise sheets in groups of two. Please give both your names on your solution, and include them in the email as well.