Automated Planning Course

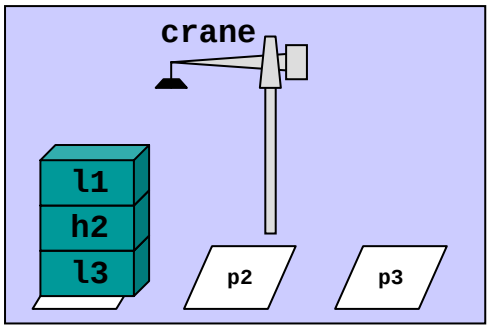
Assessed Assignment 2

This is the second of two assessed assignments for the Automated Planning course 2010/11. This assignment accounts for 15% of the final mark of the course.

Date handed out: 5/11/2010
To be handed in at the ITO: 26/11/2010, before 4pm!

1. STN/HTN Planning

Suppose the containers in the DWR domain are either heavy or light containers. Ships must be loaded such that the weight distribution is even. To be able to do this more quickly, containers are to be sorted into heavy and light piles. For example, the figure below shows a location where there is one crane, three piles, and three containers. All containers are in the same pile. The light containers are “l1” and “l3”, whereas “h2” is a heavy container.



a. Define a set of STN methods that will perform the task of sorting one given mixed pile into two piles containing only either light or heavy containers each.
b. For the initial state given in the figure above, give a solution plan and show that it accomplishes the task of sorting the stack (according to the definition of an STN solution).
c. Translate your STN methods into HTN methods. What are the most important differences between your two representations?
d. Define a set of state variables that could be used to describe world states for this problem. What would the representation of the state in the figure above look like?

2. Graphplan

The Blocks World can be described in a number of ways. Consider a representation in which there are three operators:

- Move a block from the table onto a block
- Move a block from a block onto the table
- Move a block from a block onto another block

a. Define the Sussman anomaly (see the planning book) as a propositional planning problem. Hint: This will require defining the propositional planning domain.
b. Choose a pair of independent actions that are applicable in the initial state of the Sussman anomaly. What is the result of applying these actions? If there are no such actions, show that there is no such pair.
c. Develop the planning graph for this problem far enough to be able to show an example of a pair of mutex actions that are not dependent. If there are no such actions, stop at proposition layer P_2. Why are there never any mutex but independent actions in the first action layer A_1?
d. What is the time complexity for expanding the planning graph with a new action and a new proposition layer?

MSc only question

Graphplan turned out to be significantly faster than any of the algorithms that came before it (state-space search and plan-space search; HTN planners solve a different problem). Why do you think that is?