

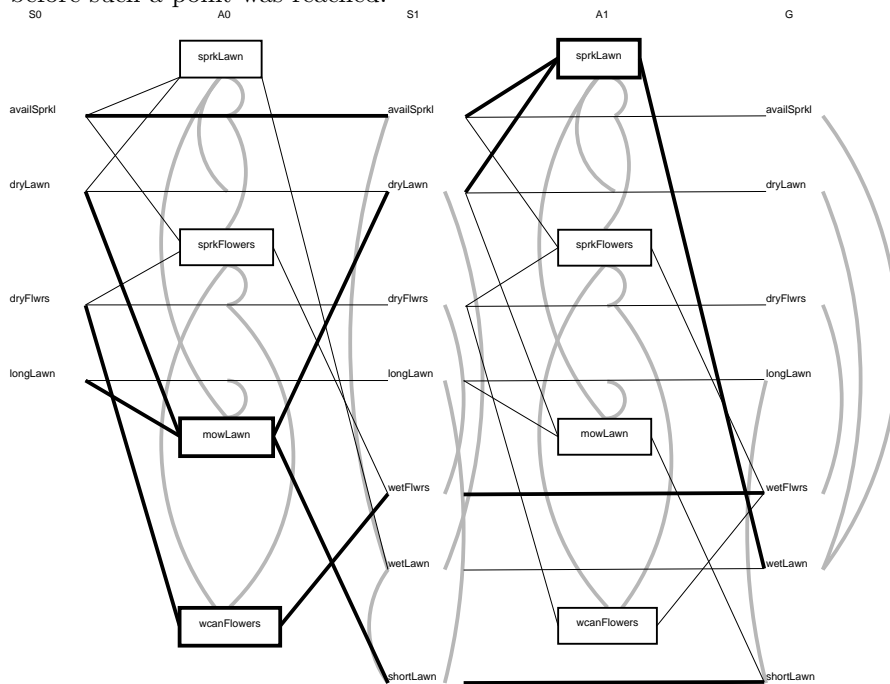
TDT4136, Assignment 6

Kim Rune Solstad

November 20, 2013

1 Planning Graph

Negations has been ignored in this graph, as most of the conditions will function as negations. Assumptions that the agent is able to do more than one task has been made. That is why the plannin graph is non-serial. Also, the graph is not expanded until the graph has leveled-off. This is because a solution was found before such a point was reached.



1.1 S_0

The preconditions in the initial state of this planning graph allows all actions to be applied. Here none of the goal conditions has been satisfied. So the graph is expanded.

1.2 S_1

Here all goalconditions has been satisfied. There exists some mutex relations preventing all goals to be reached, the level is therefore a no-good. The most

important mutex relations are:

1. The actions *sprkLawn* and the action *mowLawn* are mutex because they both require the precondition *dryLawn*.
2. The effects *wetLawn* and *shortLawn* are mutex because *sprkLawn* and *mowLawn* are mutex.

1.3 G

In this state, the goals are not mutex. This means that we can attempt to extract a solution. Whether further expansion of the graph is needed, will be determined by the success or failure of the algorithm used to extract the solution.

2 Extraction of a solution

2.1 Path found

This is a list of actions applied in chronological order to reach the goal. The order of actions applied in a single action-level is not important.

- A_0
 - *availSprkl*
 - *mowLawn*
 - *wcanFlowers*
- A_1
 - *sprkLawn*
 - *wetFlwrs*
 - *shortLawn*

2.2 Extracting the solution

The solution is found using a backtracking algorithm. The cost for a action is 1 and the cost for a persistence action is 0. The starting state, G is at the end of the planning-graph. First the persistence-actions of *wetFlws*, *wetLawn* and *shortLawn* are selected. The *wetLawn* and *shortLawn* in S_1 is mutex. One of the persistence-actions is then reverted. The *wetLawn* persistent action is chosen randomly. The *sprkLawn* is then applied as it is the only possible option after eliminating the persistent action. The conditions reached so far, in level S_1 , are:

1. *availSprk*
2. *dryLawn*
3. *wetFlwrs*
4. *shortLawn*

The persistence actions of *availSprkl* and *dryLawn* is chosen. The *sprkFlowers* action is mutex with the persistence action of *availSprkl*, therefore the *wcanFlowers* is chosen instead. Lastly the *mowLawn* action is selected for the *shortLawn* condition, wich is the only action available. The initial state has now been reached. The conditions met are:

1. *availSprkl*
2. *dryLawn*
3. *dryFlwrs*
4. *longLawn*

Since the initial conditions have been met, the extracting algorithm has been sucessfull. There is no need for further expansion of the graph.