Homework Assignment for Chapter 11, Chapter 13, and Sections 14.1–14.3
CMSC 421, Spring 2010

- This assignment is due April 20.
- There are 6 problems worth 10 points each, for a total of 60 points.

Chapter 11: Ignore the chapter, and use my lecture slides on planning instead. Do the planning homework given below.

Chapter 13: Do problems 13.10 and 13.12 from Russell & Norvig.

Sections 14.1–14.3: Do problems 14.1 and 14.7(a,b,c)

Planning Homework: Suppose you have a coffee maker and several different kinds of coffee beans, and the coffee maker can be used to make pots of coffee. There are three operators: load coffee into the machine, brew coffee, and unload the machine:

load(x)
  precond: coffee(x), loaded(nil)
  effects: loaded(x), ¬loaded(nil)

unload(x)
  precond: loaded(x), ¬loaded(nil)
  effects: ¬loaded(x), loaded(nil)

brew(x)
  precond: loaded(x), ¬loaded(nil), ¬loaded(waste)
  effects: ¬loaded(x), loaded(waste), pot(x)

Problem 1. Suppose we run Graphplan with the following the initial state and goal formula:

Initial state: {coffee(caf), coffee(decaf), loaded(nil)}

Goal formula: {pot(caf), pot(decaf)}

(a) Draw the planning graph after the first graph-expansion. Include all maintenance actions and mutexes. For each mutex, tell what kind of mutex it is.

(b) At what level does Graphplan first call Extract?

Problem 2. To encode the above problem as a task-list planning problem, we can use the operators and the initial state given above, along with a single method:

make(x, y, z)
  Task: pot(x)
  Precond: *none*
  Subtasks: ⟨load(x), brew(y), unload(z)⟩

(a) What solutions can TFD find if the initial task list is ⟨pot(caf),pot(y)⟩?

(b) Add an additional method to enable TFD to find optimal (i.e., shortest) solutions to coffee-making problems.