advanced-planning-exercises

March 26, 2019

1 11. Planning and Acting in the Real World

11.1 The goals we have considered so far all ask the planner to make the world satisfy the goal at just one time step. Not all goals can be expressed this way: you do not achieve the goal of suspending a chandelier above the ground by throwing it in the air. More seriously, you wouldn't want your spacecraft life-support system to supply oxygen one day but not the next. A *maintenance goal* is achieved when the agent's plan causes a condition to hold continuously from a given state onward. Describe how to extend the formalism of this chapter to support maintenance goals.

11.2 You have a number of trucks with which to deliver a set of packages. Each package starts at some location on a grid map, and has a destination somewhere else. Each truck is directly controlled by moving forward and turning. Construct a hierarchy of high-level actions for this problem. What knowledge about the solution does your hierarchy encode?

11.3 [HLA-unique-exercise] Suppose that a high-level action has exactly one implementation as a sequence of primitive actions. Give an algorithm for computing its preconditions and effects, given the complete refinement hierarchy and schemas for the primitive actions.

11.4 Suppose that the optimistic reachable set of a high-level plan is a superset of the goal set; can anything be concluded about whether the plan achieves the goal? What if the pessimistic reachable set doesn't intersect the goal set? Explain.

11.5 [HLA-progression-exercise] Write an algorithm that takes an initial state (specified by a set of propositional literals) and a sequence of HLAs (each defined by preconditions and angelic specifications of optimistic and pessimistic reachable sets) and computes optimistic and pessimistic descriptions of the reachable set of the sequence.

11.6 In Figure Section **??** we showed how to describe actions in a scheduling problem by using separate fields for , , and . Now suppose we wanted to combine scheduling with nondeterministic planning, which requires nondeterministic and conditional effects. Consider each of the three fields and explain if they should remain separate fields, or if they should become effects of the action. Give an example for each of the three.

11.7 Some of the operations in standard programming languages can be modeled as actions that change the state of the world. For example, the assignment operation changes the contents of a memory location, and the print operation changes the state of the output stream. A program consisting of these operations can also be considered as a plan, whose goal is given by the specification of the program. Therefore, planning algorithms can be used to construct programs that achieve a given specification.

1. Write an action schema for the assignment operator (assigning the value of one variable to another). Remember that the original value will be overwritten!

2. Show how object creation can be used by a planner to produce a plan for exchanging the values of two variables by using a temporary variable.

11.8 Consider the following argument: In a framework that allows uncertain initial states, **nondeterministic effects** are just a notational convenience, not a source of additional representational power. For any action schema a with nondeterministic effect $P \vee Q$, we could always replace it with the conditional effects $R: P \wedge \neg R: Q$, which in turn can be reduced to two regular actions. The proposition R stands for a random proposition that is unknown in the initial state and for which there are no sensing actions. Is this argument correct? Consider separately two cases, one in which only one instance of action schema a is in the plan, the other in which more than one instance is.

11.9 [conformant-flip-literal-exercise] Suppose the $Flip$ action always changes the truth value of variable L. Show how to define its effects by using an action schema with conditional effects. Show that, despite the use of conditional effects, a 1-CNF belief state representation remains in 1-CNF after a $Flip$.

11.10 In the blocks world we were forced to introduce two action schemas, *Move* and *MoveToTable*, in order to maintain the *Clear* predicate properly. Show how conditional effects can be used to represent both of these cases with a single action.

11.11 [alt-vacuum-exercise] Conditional effects were illustrated for the *Suck* action in the vacuum world—which square becomes clean depends on which square the robot is in. Can you think of a new set of propositional variables to define states of the vacuum world, such that *Suck* has an *unconditional* description? Write out the descriptions of *Suck*, $Left$, and *Right*, using your propositions, and demonstrate that they suffice to describe all possible states of the world.

11.12 Find a suitably dirty carpet, free of obstacles, and vacuum it. Draw the path taken by the vacuum cleaner as accurately as you can. Explain it, with reference to the forms of planning discussed in this chapter.

11.13 The following quotes are from the backs of shampoo bottles. Identify each as an unconditional, conditional, or execution-monitoring plan. (a) "Lather. Rinse. Repeat." (b) "Apply shampoo to scalp and let it remain for several minutes. Rinse and repeat if necessary." (c) "See a doctor if problems persist."

11.14 Consider the following problem: A patient arrives at the doctor's office with symptoms that could have been caused either by dehydration or by disease D (but not both). There are two possible actions: *Drink*, which unconditionally cures dehydration, and *Medicate*, which cures disease D but has an undesirable side effect if taken when the patient is dehydrated. Write the problem description, and diagram a sensorless plan that solves the problem, enumerating all relevant possible worlds.

11.15 To the medication problem in the previous exercise, add a *Test* action that has the conditional effect $CultureGrowth$ when *Disease* is true and in any case has the perceptual effect $Known(CultureGrowth)$. Diagram a conditional plan that solves the problem and minimizes the use of the *Medicate* action.