Automated Planning Course

Assessed Assignment 1

This is the first 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: 13/10/2010
To be handed in at the ITO: Wednesday, 3/11/2010, before 4pm!

1. **State Transition Systems**: Define a state transition system for a ticket vending machine that accepts 10p, 20p, and 50p coins; the price of a ticket is 60p. Users must insert coins until at least 60p have been inserted. The machine then issues a ticket and returns correct change.

2. **Situation Calculus**: Go to the course book web-site (http://www.laas.fr/planning/) and download the PDDL specification of the DWR domain defined there. Define the `move` operator as a set of applicability, effect, and frame axioms in the style of the situation calculus.

3. **State-Space Search**:
   a. Go to the course book web-site (http://www.laas.fr/planning/) and download the PDDL specification of the two problems defined there. Draw the initial state for the smaller problem (1 robot/2 locations) as a picture. This should show the topology and location of different objects defined in the problem. Draw a possible goal state for the smaller problem. How many different goal states are there?
   b. Some operators may have instances with effects that are complementary, i.e. they contain effects `(P …)` and `(not P …)`. Show an example of such an operator in the DWR domain. What would happen if we used such an action in the DWR domain? Why is this not an issue in the two example problems?
   c. What are the applicable actions in the initial state of the smaller DWR problem? What are the applicable actions in the (random) planning problem given in the appendix? Ignore actions with inconsistent (complementary) effects.
   d. For the same two problems, which actions are relevant for the respective goals? Note that static relations can be used to constrain the search for relevant actions, and type information where available. Again, ignore actions with inconsistent (complementary) effects.

4. **Plan-Space Search**:
   a. Define a partial plan that represents the initial state for the 1 robot/2 locations problem in the plan-space search approach.
   b. Simulate the search as performed by the PoP procedure for the first 3 levels. When there is a choice point, choose one option to simulate, but indicate which choice points are deterministic and which are non-deterministic, i.e. which ones are backtrack points.

MSc only question:

a. The algorithm for finding applicable actions described in the lecture does not specify the order in which preconditions should be considered. Suggest some useful heuristics.
b. Discuss the differences (advantages and disadvantages) between state- and plan-space search.

Appendix: Random Planning Domain and Problem

```
(define (domain random-domain)
  (:requirements :strips)
  (:action op1
    :parameters (?x1 ?x2 ?x3)
    :precondition (and (Q ?x2) (Q ?x1) (R ?x3 ?x3) (R ?x1 ?x1)
      (not (P ?x3 ?x1 ?x3)) (not (S ?x3 ?x1)))
    :effect (and (R ?x1 ?x2) (S ?x1 ?x2) (S ?x2 ?x3) (S ?x1 ?x3)
      (not (R ?x3 ?x3)) (not (R ?x1 ?x1))))
  (:action op2
    :parameters (?x1 ?x2 ?x3)
    :precondition (and (Q ?x3) (R ?x3 ?x2) (P ?x3 ?x1 ?x2) (P ?x3 ?x3 ?x1)
      (not (T ?x1 ?x3 ?x3)) (not (T ?x1 ?x2 ?x3)))
    :effect (and (T ?x2 ?x1 ?x3) (S ?x3 ?x3) (R ?x3 ?x1) (R ?x3 ?x3)
      (not (R ?x3 ?x2))))
  (:action op3
    :parameters (?x1 ?x2 ?x3)
    :precondition (and (T ?x2 ?x3 ?x2) (S ?x2 ?x1) (T ?x1 ?x1 ?x2) (Q ?x1)
      (not (T ?x2 ?x3 ?x1)) (not (S ?x2 ?x3)) (not (R ?x1 ?x3)))
    :effect (and (R ?x2 ?x2) (R ?x1 ?x2) (S ?x2 ?x2) (S ?x2 ?x3)
      (not (S ?x2 ?x1)) (not (T ?x1 ?x1 ?x2)))))


(define (problem random-problem)
  (:domain random-domain)
  (:init
     (Q I) (Q G) (Q H) (Q D) (Q J)
     (P C I G) (P J A E) (P J I E) (P A D C) (P I C C) (P G D C) (P G E I)
      (P B E A) (P B G E) (P I B J) (P I F E) (P C E F) (P G I H) (P J G C)
      (P A A J)
     (T C A F) (T H C E) (T H B I) (T A F H) (T G C J) (T J H A)
     (S D E) (S B J) (S G J) (S J F) (S D G) (S A I) (S A B) (S D I)
     (R I I) (R B C) (R D C) (R I C) (R I F) (R C I))
  (:goal (and (S I G) (S J I))))
```