15-889 Fall 1997 – Final Exam

Due: Monday, December 15, 1997

Problem 1

There are many possible ways to constrain resource selection (aka variable bindings, variable instantiation and separation, etc) in a planning algorithm. Three ways are: (i) use-when conditions (constraining functions); (ii) the use of static preconditions (static preconditions refer to predicates that cannot change); and (iii) control rules.

Compare these methods in terms of:

(a) semantics;

(b) planning efficiency;

(c) planning completeness.

Justify your answers.

Problem 2

One of the steps in the UCPOP planning algorithm is the identification of flaws. Flaws can be open goals or threats to resolve. UCPOP select one of flaw and does not keep the other flaws as backtracking points. Could this be a cause of planning incompleteness for UCPOP? Prove your answer.

Problem 3

Consider the operator to load an object into a briefcase in the briefcase domain (see page 52 of Weld'94):

```
Put-in
  pre: (and (briefcase ?b)
            (diff ?x ?b)
            (at ?x ?l)
            (at ?b ?l))
  add: (in ?x ?b)
  del: --
```

Consider the problem in the same article, as stated on pages 51–53. Use the move and take-out operators. Discuss the effect of slightly changing the load operator to be:

```
Put-in
  pre: (and (briefcase ?b)
            (diff ?x ?b)
            (at ?x ?l)
            (at ?b ?l))
  add: (in ?x ?b)
  del: (at ?x ?l)
```

Problem 4

Let the length of a plan be the total number of steps in the plan, independently from their orderings.

(a) Show an example where Graphplan would find a plan with N steps while Prodigy (or UCPOP) would find a plan of 2 steps.

(b) Show an example where Prodigy (or UCPOP) would find a plan with N steps while Graphplan would find a plan of 2 steps.

(c) Would you be able to answer the questions above for 1 step instead of 2 steps? How or why not?

Problem 5

Consider the following three domains: 8-puzzle, water jug, and logistics (see the short descriptions below). Discuss which planning technique, namely backward-chaining, HTN planning, Graphplan, or Satplan would be more appropriate for each of domain.

In the 8-puzzle domain, there are 9 titles, one of which is blank and the others are numbered 1 through 8. The titles can slide and a planning instance consists of finding the sequence of title sliding movements to achieve a final configuration.

In the water jug domain, several jugs are given with different capacities. There is a pump that can be used to fill the jugs with water to their full capacity. The water of one jug can be poured into other jugs also up to the full capacity of the receiving jug. Any amount of water in a jug can be dumped.

The logistics domain involves movement of packages between different locations using different carriers.

Problem 6

One classical approach to planning is least commitment, partial order planning. Another approach is to make early commitments and total order planning. For instance, during planning additional bindings and/or temporal orderings might be added in order to make the plan more concrete, and access the effects of the plan in a resulting state.

(a) Consider the use of a transformational planner together with early commitment and total order planning. Discuss the possible tradeoffs in terms of search efficiency (both the number of nodes searched and the amount of computation to expand and evaluate each node) between this approach and the least commitment, partial order approach. Present at least one argument in favor of each approach.

(b) Consider the Saba heuristic used in Prodigy which postpones ordering commitments until all the goals are planned for and there is a fringe of steps all linked to the initial state. Is the Saba approach equivalent to least commitment, partial order planning? Why or why not?

Problem 7

Consider the following two operations in the Blocks World:

```
unstack (x,y)                  putdown(x)
 preconds: (on x y)             preconds: (holding x)
           (clear x)            adds: (on-table x)
           (arm-emtpy)                (clear x)
 adds: (holding x)                    (arm-empty)
       (clear y)                dels: (holding x)
 dels: (on x y)
       (clear x)
       (arm-empty)
```

Suppose that the goal is (clear B) and the initial state is as follows:

```
(on A B)                   ---
(on-table B)              | A |
(on-table C)         ---   ---
(clear A)           | C | | B |
(clear C)         ===============
(arm-empty)
```

(a) Show the search space explored by backward chaining from the goal.

(b) Use this example to show how EBL would learn an operation for achieving the goal (clear <x>) and the preconditions of this operation. Present the result of learning in the form of an if-then-else rule that describes a situation from which the target goal, (clear <x>), can be achieved.

(c) Now consider the goal (clear C) and the following new initial state:

```
            ---
           | A |
            ---
           | B |
     ---    ---
    | D |  | C |
   ===============
```

Show that the learned rule does not help in this problem. Why? What conclusion about EBL learning and the precise match reuse process can you draw from this example?

(d) Suppose that instead of learning a control rule, the planning episode is recorded in a plan library. Show how this past plan can be used to plan by analogy and how it can help reduce the search in our last problem.

(e) Analyze the general problem of recursive explanations in the context of explanation-based learning, and analogical reasoning.

Problem 8

Some of the conditional planners we studied distinguish observational effects from "causal" effects, while others do not. For instance, an operator might distinguish between seeing whether a block has a certain color *vs.* trying to paint it that color. Discuss the problems that may arise from not maintaining such a distinction.

Problem 9

Since optimal planning with POMDPs is highly intractable, some planners instead first solve the underlying MDP, and *then* add in state uncertainty. Approaches that have appeared in the literature include choosing:

1. the action associated with the most likely state,
2. the action with the highest probability mass, and
3. the action whose "Q" values, summed over the state distribution, is maximal.

(a) Explain the tradeoffs between the approaches, by identifying example domains where each approach has advantages over the others.

(b) Present an example domain where solving the POMDP directly leads to plans that are significantly better than what the stated approaches can do.

Problem 10

In three-layered architectures, typically one of the layers is "in control." For example:

- In the Animate Agent Architecture, the sequencer (RAP) is in control, activating skills and invoking the planner when needed;
- In Rogue, the planner is in control – it decides when to send actions to the sequencer (TCA);
- In CIRCA, the real-time system is basically in control – once the AI system sends it a schedule, it is largely autonomous.

Each of these approaches has advantages and disadvantages in terms of architectural complexity, modularity, concurrency, reactivity and maintenance of redundant information.

Discuss which approach may be more appropriate for the following three domains:

(a) Office delivery domain.
(b) Planetary rover navigation.
(c) Space craft scheduling.