Exam for IN4010TU Artificial Intelligence Techniques

7 November 2012

This exam will test your knowledge and understanding of the material discussed in the first period of the course Artificial Intelligence Techniques. Using the book, lecture notes, or slides during the examination is *not* allowed. You will have 3 hours (from 14 till 17) to complete the exam. It has 4 questions, for a total of 74 points. Please don't include irrelevant information: you will be marked down for this. Before you hand in your answers, please check that you have put your name and student number on top of every sheet you hand in.

Questions

Question 1 *12 points*

This question concerns some definitions.

(a) (4 points) Categorize the task environment of the game of outdoor, one-on-one tennis with respect to the following categories: fully observable vs. partially observable; single agent vs. multiagent; deterministic vs. stochastic; episodic vs. sequential; static vs. dynamic; discrete vs. continuous.

(b) (3 points) Define what skolemization is. Explain in your answer when to use skolem functions, rather than skolem constants.

(c) (3 points) Define what the closed-world assumption is. Would you use the closed-world assumption or the open-world assumption in a partially observable environment?

(d) (2 points) Randomized behavior can sometimes be rational. Provide a reason to incorporate random behavior into the design of an agent, and provide an example.

Question 2 *25 points*

This question is about agent programs, Prolog, and GOAL.

Consider the agent program listed in Figure 1. The agent's task is to manage books in a library. Books can be in one of four states:

1. they are loaned to members, indicated by `status(Book, Person, Date)` where `Date` is of the form `yy/mm/dd` where `yy` refers to the year, `mm` refers to the month, and `dd` refers to the day that the book should be returned.
2. they have been returned but still need to be put back on a shelf, indicated by `status(Book, tbs)`.
3. they are on a shelf in a book rack, indicated by `status(Book, shelf)`.
4. a fine has been issued for a book that has been returned too late, indicated by `status(Book, Person, fine)`.

Books can only be loaned to members and only if they are on the shelf. A return date is determined by adding 14 days to the current date. We assume that the predicate `add/7` provides the correct return date by adding a specified number of days (14 in our case) to the current date. A predicate of the form `status(Book, Person, YY/MM/DD)` indicates that a book has been loaned and should be returned at a particular date. *In your answers to the questions, do not introduce new predicates that have not been mentioned either in the program or this question.*

(a) (5 points) Provide a definition for the predicate `tooLate`. Your definition should capture the intuition that a person is returning a book too late if the current date is strictly later than the return date associated with the book. Your definition should work for all possible dates. You may use (in)equality signs such as < and >=.

(b) (4 points) Complete the action specification for the action `handOut`.

(c) (6 points) Complete the action specification for the action `issueFine`. How often will a person be fined given your specification? Explain in detail how you arrived at your conclusion.

(d) (4 points) List all (built-in or other) actions that can be selected by means of the rules in the **main** module and executed by the GOAL agent given the agent's *current* beliefs and goals that are listed in Figure 1.

(e) (6 points) There is a problem with the agent program. Explain in detail what goes wrong by identifying 2 issues, and propose a solution.

```
init module{
  knowledge{
    returnDate(Book, YY, MM, DD) :- status(Book, shelf), date(YYnow/MMnow/DDnow),
        add(YY, MM, DD, YYnow, MMnow, DDnow, 14).
    tooLate(Person) :- ... .
  }
  beliefs{
    member(john). member(claire). member(patricia).
    status(book1, claire, 12/10/26). status(book2, shelf). status(book3, john, 12/11/11).
    status(book4, tbs). status(book5, patricia, 12/11/13). status(book6, tbs).
    date(12/11/07).
  }
  goals{ status(book4, shelf). }
  actionspec{
    handOut(Book, Person){
      pre{ ... }
      post{ ... } }
    takeIn(Book){
      pre{ status(Book, YY/MM/DD) }
      post{ status(Book, tbs), not(status(Book, YY/MM/DD)) } }
    putOnShelf(Book){ % put book back on shelf
      pre{ status(Book, tbs) }
      post{ status(Book, shelf) }
    issueFine(Person){
      pre{ ... }
      post{ ... } }
  }
}
main module{
  program{
    if bel(status(Book, tbs)) then adopt(status(Book, shelf)).
    if a-goal(status(Book, shelf)) then putOnShelf(Book).
    if bel(request(Book)) then handOut(Book).
    if bel(returned(Book)) then takeIn(Book).
    if bel(tooLate(Person)) then issueFine(Person).
  }
}
```

Figure 1: GOAL Agent Program

Question 3 *17 points*

Consider the following planning problem that involves a care robot working in a hospital. The robot can move between a patient room and a preparation room, and is capable of picking up only one object at a time.

```
Init(emptyGripper ∧ medicineInCabinet ∧ inPreparationRoom)

Goal(patientHasMedicine ∧ patientHasWater)

Action(pickUpMedicine,
PRECOND: emptyGripper ∧ medicineInCabinet ∧ inPreparationRoom
EFFECT: ¬medicineInCabinet ∧ ¬emptyGripper ∧ holdingMedicine)

Action(fillGlass,
PRECOND: emptyGripper
EFFECT: waterInGlass)

Action(pickUpWater,
PRECOND: emptyGripper ∧ waterInGlass ∧ inPreparationRoom
EFFECT: ¬emptyGripper ∧ holdingWater)

Action(gotoPatient,
PRECOND: inPreparationRoom
EFFECT: inPatientRoom ∧ ¬inPreparationRoom)

Action(gotoPreparationRoom,
PRECOND: inPatientRoom
EFFECT: inPreparationRoom ∧ ¬inPatientRoom)

Action(givePatientMedicine,
PRECOND: inPatientRoom ∧ holdingMedicine
EFFECT: ¬holdingMedicine ∧ patientHasMedicine ∧ emptyGripper)

Action(givePatientWater,
PRECOND: inPatientRoom ∧ holdingWater
EFFECT: ¬holdingWater ∧ patientHasWater ∧ emptyGripper)
```

(a) (15 points) Construct a solution for this planning problem using partial order planning. (Tip: first think of a solution yourself, and consider where conflicts could arise!)

(b) (2 points) How many linearizations does your partial order plan have?

Question 4 *20 points*

This question is about reinforcement learning. Consider the state transition diagram in Figure 2. Assume a "greedy policy", i.e. action selection chooses the best possible action, $\gamma = 0.7$, and that learning has converged (is finished). (Hint: remember that in Reinforcement Learning the action selection mechanism influences the propagated values as the values must converge to the Bellman equation.)

(a) (15 points) Compute the value $v(s)$ for every state s.

(b) (5 points) Explain which action the agent should perform in the state "In car".

End of exam

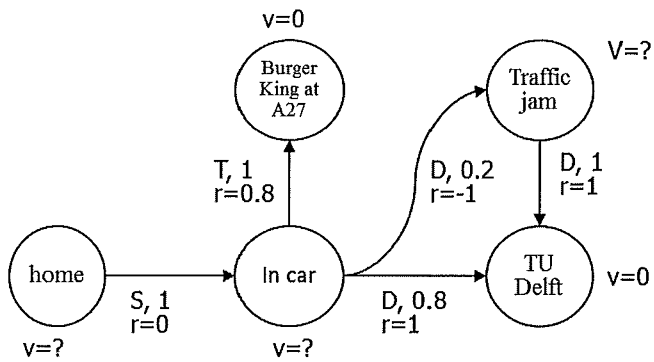

Figure 2: State Transition Diagram