CPSC 422

Practice Final Exam

April 2011

This practice exam is meant to give some example questions. You can expect questions that show your knowledge of the details (e.g., asynchronous value iteration, Q-learning), as well as testing your understanding about the bigger picture. Note that this practice exam emphasizes the last half of the course. The final exam will cover all parts of the course. You should also look at the practice midterm and the actual midterm (and their solutions). There will be similar questions to the midterm (particularly the questions students did not do as well at).

You may bring into the exam a letter-sized sheet of paper with anything written on it. You may not use calculators, PDAs, robotic assistants or other electronic aids.

Some important points (that students often forget):

- Read and answer the question. You will not get marks for writing things (whether they are true or not) that are not relevant to the question.
- Use proper English in full sentences. You will not get marks if we cannot work out what you are saying.
- If a question asks about a particular instance of a problem, make sure your answer refers to that instance. Writing a general formula that you may have copied from the sheet you can bring in, is not worth any marks. (The questions are usually asking to apply that formula to a particular case, to make sure you understand it).

The course covered the following sections of the textbook:

- Agents: Sections 1.3, 1.5, 2.1, 2.2, 2.3
- Probability and time: Section 6.5
- Probabilistic inference: Section 6.4
- Learning: Sections 7.1-7.4, 7.8, 11.1, 11.2

- Decision making: Chapter 9 and Section 11.3
- Individuals and relations: Section 13.1-13.3, 14.3. [Note that you will not be expected to memorize any of OWL; we will give any constructs you need.]

If you can do the questions below, you should have no trouble with the final exam. Note that you will not be expected to remember the syntax or semantics of OWL or of the independent choice logic, but you should be able to read it.

1. Reinforcement Learning
   (a) In the "simple game" with treasures that appear in the corners, there are features that are the x-distance the current treasure and features that are the y-distance to the current treasure. Chris thought that these were not useful as they don't depend on the action. However, when she removed them she found that the controller performed very poorly. Explain to Chris how these features help.
   (b) Why do people use Q-learning and SARSA for games, but don't use them for robots? What else can be used for robots? Explain why.
   (c) For a model-based reinforcement learner that reasons in terms of individual states with 1000 states and 4 actions at every state, what data structures are required and what is their size?
   (d) Which of the following reinforcement algorithms will find the optimal policy, given enough time. Which ones will actually follow the optimal policy? Explain why.
      i) Q-learning with fixed α and 80% exploitation.
      ii) Q-learning with fixed $\alpha_k = 1/k$ and 80% exploitation.
      iii) Q-learning with $\alpha_k = 1/k$ and 100% exploitation.
      iv) SARSA-learning fixed $\alpha_k = 1/k$ and 80% exploitation.
      v) SARSA-learning fixed $\alpha_k = 1/k$ and 100% exploitation.
      vi) Feature-based SARSA learning with soft-max action selection.
      vii) A model based-reinforcement learner with 50% exploitation.
2. A travel site has a Prolog database that represents information about hotels and feedback from users that uses the relations:

   $$hotel(Id, Name, City, Country, Address)$$

   $$reported\_clean(Hotel, RoomNumber, Cleanliness, day(Year, Month, Day))$$

Show how the following facts can be represented using triple notation, using vocabulary that make sense:

```
hotel(h345,"The Beach Hotel",adelaide,australia,"300 Beach St").
reported_clean(h345,127,clean,day(2011,01,25)).
```

Show how this information can be represented using individual-property-value triples.

Is it reasonable to represent the hotel name and address as strings? Explain.

3. Consider the following ontology (written in the OWL functional syntax) about hotels and hotel rooms. In this ontology, a hotel has rentable units, each of which is a suite, a standard room or a room.

```
Declaration(Class(:Hotel))
Declaration(Class(:Room))
Declaration(Class(:BathRoom))
SubClassOf(:BathRoom :Room)

Declaration(Class(:RentableHotelUnit))
Declaration(Class(:RoomOnly))
SubClassOf(:RoomOnly :RentableHotelUnit)
SubClassOf(:RoomOnly :Room)
Declaration(Class(:StandardRoom))
EquivalentClasses(:StandardRoom
        ObjectIntersectionOf(
             :RentableHotelUnit
             ObjectSomeValuesFrom(:ContainsRoom :BathRoom)
             ObjectExactCardinality(2 :ContainsRoom :Room)))
Declaration(Class(:Suite))
EquivalentClasses(:Suite
        ObjectIntersectionOf(
             :RentableHotelUnit
             ObjectSomeValuesFrom(:ContainsRoom :BathRoom)
             ObjectMinCardinality(3 :ContainsRoom :Room)))

Declaration(ObjectProperty(:ContainsRoom))
InverseFunctionalObjectProperty(:ContainsRoom)
ObjectPropertyDomain(:ContainsRoom :RentableHotelUnit)
ObjectPropertyRange(:ContainsRoom :Room)

Declaration(ObjectProperty(:HasForRent))
InverseFunctionalObjectProperty(:HasForRent)
ObjectPropertyDomain(:HasForRent :Hotel)
```

```
ObjectPropertyRange(:HasForRent :RentableHotelUnit)

Declaration(Class(:XX))
EquivalentClasses(:XX
        ObjectIntersectionOf(
             ObjectSomeValuesFrom(:HasForRent :Suite)
             :Hotel))
Declaration(Class(:YY))
EquivalentClasses(:YY
        ObjectIntersectionOf(
             ObjectMaxCardinality(0 :HasForRent :Suite)
             :Hotel))
Declaration(Class(:ZZ))
EquivalentClasses(:ZZ
        ObjectIntersectionOf(
             ObjectAllValuesFrom(:HasForRent :Suite)
             :Hotel))
```

(a) Explain in English what a standard room is.

(b) Is a standard room a room? Should it always be? Should it never be? Explain.

(c) Can a standard room contain two bathrooms? How could the ontology be modified to disallow this?

(d) In this ontology, we have replaced some of the classes with *XX*, *YY*, and *ZZ*. For each of the following, state which of *XX*, *YY* or *ZZ* defines this concept:

i) an "all suites hotel"

ii) a hotel with some suites

iii) a hotel with no suites

4. Suppose we have parmetrized random variables $likes(Person, Movie)$, $young(Person)$ and $genre(Movie, Genre)$ where there at 1000 people, 100 movies and 5 genres. Suppose $young(Person)$ and $genre(Movie, Genre)$ are parents of $likes(Person, Movie)$.

(a) Draw this in plate notation.

(b) How many random variables are in the grounding of this model?

(c) Draw the grounding belief network assuming the population of *Person* is $\{sam, chris, kim\}$ and the population of *Movie* is $\{terminator, rango\}$.

(d) Consider the following fragment of independent choice logic theory:

```
likes(Person,Movie) <- young(Person) & genre(Movie,action) &
     young_likes_action(Person,Movie).
prob young_likes_action(Person,Movie):0.7.
likes(Person,Movie) <- ~young(Person) & genre(Movie,action) &
     old_likes_action(Person,Movie).
prob old_likes_action(Person,Movie):0.6.
likes(Person,Movie) <- genre(Movie,comedy) &
     likes_comedy(Person,Movie).
prob likes_comedy(Person,Movie):0.8.
young(sam).
young(chris).
genre(terminator,action).
genre(rango,comedy).
```

where `<-` means "if", `~` means "not", `&` means "and", `prob a:p.` means $\{a, \tilde{}a\}$ is an alternative with $P(a) = p$.

i) What is the probability of $likes(sam, rango)$?

ii) What is the probability of $likes(chris, terminator)$?

iii) What is the probability of $likes(kim, terminator)$?