CS 188	Introduction to AI	
Fall 2002	Stuart Russell	Final

You have 2 hours and 50 minutes. The exam is open-book, open-notes. 100 points total. Panic not.

ALL QUESTIONS IN THIS EXAM ARE TRUE/FALSE, MULTIPLE-CHOICE, OR SHORT-ANSWER.
Mark your answers on the exam itself and turn it in. Write your name and SID at the bottom of each page
For true/false questions, CIRCLE *True* OR *False*.
For multiple-choice questions, CIRCLE *ALL* CORRECT CHOICES (in some cases, there may be more than one).

If you are not sure of your answer you may wish to provide a *brief* explanation. Don't waste time doing this if your answer is correct!

DO NOT DISCLOSE ANY EXAM CONTENT OR DISCUSS WITH OTHER STUDENTS!!!!!

For official use only

Q. 1	Q. 2	Q. 3	Q. 4	Q. 5	Q. 6	Q. 7	Q. 8	Q. 9	Total
/10	/13	/6	/10	/8	/16	/12	/15	/10	/100

1. (10 pts.) Some Easy Questions to Start With

(a) (2) *True/False*: Ambiguity in natural language can always be resolved by checking dictionary definitions of words.

(b) (2) *True/False*: An agent that communicates using natural language is usually operating in a partially observable environment.

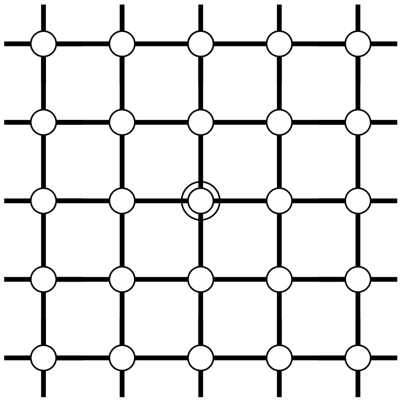
(c) (2) *True/False*: A feedforward neural network of sufficient size can generate successful behavior in most partially observable environments.

(d) (2) *True/False*: A computer with a trillion terabytes of memory capable of a trillion teraflops would be smarter than a human being.

(e) (2) *True/False*: Using odometry, a mobile robot can keep track of its location for an unbounded time.

2. (13 pts.) Search

Consider the unbounded regular 2D grid state space shown below. The start state is the origin (marked) and the goal state is at (x, y).



(a) (1) What is the branching factor b in this state space? ______________

(b) (2) How many distinct states are there at depth k (for $k > 0$)?
(i) 4^k (ii) $4k$ (iii) $4k^2$

(c) (2) Breadth-first search *without* repeated-state checking *expands* at most
(i) $((4^{x+y+1} - 1)/3) - 1$ (ii) $4(x + y) - 1$ (iii) $2(x + y)(x + y + 1) - 1$
nodes before terminating.

(d) (2) Breadth-first search *with* repeated-state checking *expands* up to
(i) $((4^{x+y+1} - 1)/3) - 1$ (ii) $4(x + y) - 1$ (iii) $2(x + y)(x + y + 1) - 1$
nodes before terminating.

(e) (2) *True/False*: $h = |u - x| + |v - y|$ is an admissible heuristic for a state at (u, v).

(f) (2) *True/False*: A* search *with* repeated-state checking using h expands $O(x + y)$ nodes before terminating.

(g) (1) *True/False*: h remains admissible if some links are removed.

(h) (1) *True/False*: h remains admissible if some links are added between nonadjacent states.

3. (6 pts.) Propositional Logic

(a) (1) *True/False*: $A \Leftrightarrow B \models A \vee B$.

(b) (1) *True/False*: $A \Leftrightarrow B \models \neg A \vee B$.

(c) (2) *True/False*: $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ is satisfiable.

(d) (2) *True/False*: $(A \Leftrightarrow B) \Leftrightarrow C$ has the same number of models as $(A \Leftrightarrow B)$ for any fixed set of proposition symbols that includes A, B, C.

4. (10 pts.) First-Order Logic

(a) (3) "Every cat loves its mother or father" can be translated as

i. $\forall x\ Cat(x) \Rightarrow Loves(x, Mother(x) \vee Father(x))$.
ii. $\forall x\ \neg Cat(x) \vee Loves(x, Mother(x)) \vee Loves(x, Father(x))$.
iii. $\forall x\ Cat(x) \wedge (Loves(x, Mother(x)) \vee Loves(x, Father(x)))$.

(b) (3) "Every dog who loves one of its brothers is happy" can be translated as

i. $\forall x\ Dog(x) \wedge (\exists y\ Brother(y, x) \wedge Loves(x, y)) \Rightarrow Happy(x)$.
ii. $\forall x, y\ Dog(x) \wedge Brother(y, x) \wedge Loves(x, y) \Rightarrow Happy(x)$.
iii. $\forall x\ Dog(x) \wedge Brother(F(x), x) \wedge Loves(x, F(x)) \Rightarrow Happy(x)$.

where F is a Skolem function.

(c) (4) Which of the following sentences are equivalent to a set of Horn clauses?
(a)(ii) (a)(iii) (b)(i) (b)(ii)

5. (8 pts.) Logical Inference
In this question we will consider Horn KBs, such as the following:

$$P(F(x)) \Rightarrow P(x)$$
$$Q(x) \Rightarrow P(F(x))$$
$$P(a)$$
$$Q(b)$$

where x is a variable and a and b are constants.

Let FC be a "breadth-first" forward-chaining algorithm that repeatedly adds all consequences of currently satisfied rules; let BC be a "depth-first left-to-right" backward-chaining algorithm that tries clauses in the order given in the KB.

(a) (1) *True/False*: FC will infer the literal $Q(a)$.

(b) (2) *True/False*: FC will infer the literal $P(b)$.

(c) (2) *True/False*: If FC has failed to infer a given literal, then it is not entailed by the KB.

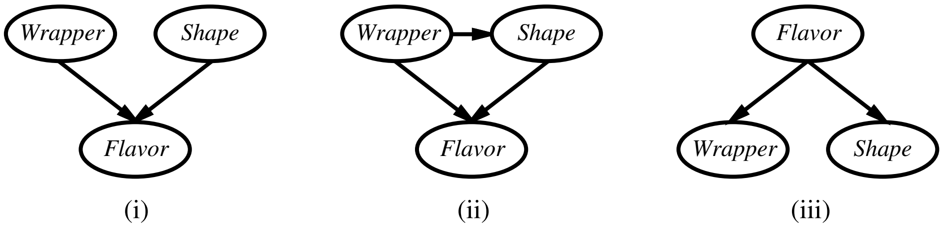
(d) (2) *True/False*: BC will return *true* given the query $P(b)$.

(e) (1) *True/False*: If BC does not return *true* given a query literal, then it is not entailed by the KB.

6. (16 pts.) Probability, Bayes Nets, Decision Theory

The Surprise Candy Company makes candy in two flavors: 70% are strawberry flavor and 30% are anchovy flavor. Each new piece of candy starts out with a round shape; as it moves along the production line, a machine randomly selects a certain percentage to be trimmed into a square; then, each piece is wrapped in a wrapper whose color is chosen randomly to be red or brown. 80% of the strawberry candies are round and 80% have a red wrapper, while 90% of the anchovy candies are square and 90% have a brown wrapper. All candies are sold individually in sealed, identical, black boxes.

Now you, the customer, have just bought a Surprise candy at the store but have not yet opened the box. Consider these three Bayes nets:



(a) (3) Which network(s) can correctly represent $P(Flavor, Wrapper, Shape)$?
(i) (ii) (iii)

(b) (1) Which network is the best representation for this problem?
(i) (ii) (iii)

(c) (1) *True/False*: Network (i) asserts that $P(Wrapper|Shape) = P(Wrapper)$.

(d) (3) What is the probability that your candy has a red wrapper?
(i) 0.8 (ii) 0.56 (iii) 0.59

(e) (3) In the box is a round candy with a red wrapper. The probability that its flavor is strawberry is
(i) ≤ 0.7 (ii) Between 0.7 and 0.99 (iii) > 0.99

(f) (2) A unwrapped strawberry candy is worth s on the open market and an unwrapped anchovy candy is worth a. Write an expression for the value of an unopened candy box.

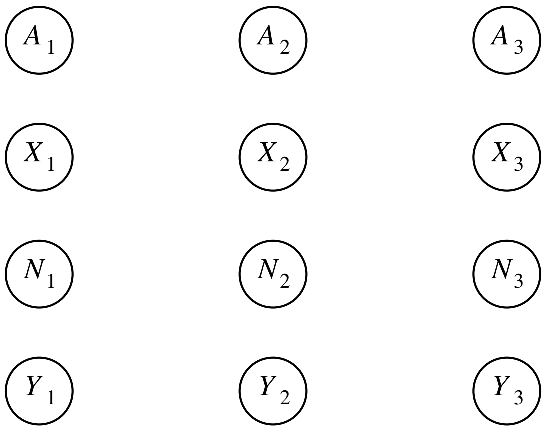
____

(g) (3) A new law prohibits trading of unwrapped candies, but it is still legal to trade wrapped candies (out of the box). How much is an unopened candy box worth now?
(i) More than before (ii) Less than before (iii) Same as before (iv) Cannot tell

7. (12 pts.) Dynamic Bayes Nets

An agent lives in a 3×3 grid world surrounded by walls. (1,1) is at bottom left and (3,1) is at bottom right. The agent's action A can be *left*, *up*, *right*, or *down*. The percept is an integer N indicating the number of adjacent walls. The agent's state is just its (X, Y) position and the prior over X and Y is uniform. At each time step the agent receives a percept and then chooses its action.

(a) (3) Here are the first three time slices of a dynamic Bayes net. Add links to make the best possible representation of the problem. (You need not add parents for A because the agent chooses its value.)



(b) (2) Suppose the wall sensor is accurate and the agent's movement is deterministic. Given $N_1 = 1$, $A_1 = right$, $N_2 = 2$, where could the agent be at time 2?

____

(c) (2) Suppose the wall sensor detects each wall with independent probability $p < 1$, but detects the absence of a wall correctly. Given $N_1 = 1$, $A_1 = right$, $N_2 = 1$, where could the agent *not* have been at time 1?

____

(d) (2) *True/False*: Given the motion model of part (b) and the sensor model of part (c), it is impossible for the agent ever to know *exactly* where it is with certainty.

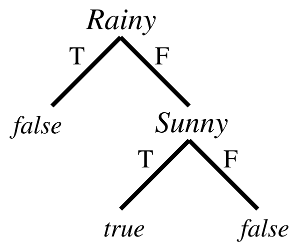
(e) (3) *True/False*: Considering only *tabular* CPTs, converting your DBN from part (a) into an HMM would increase the number of parameters needed to specify the model.

NAME:__________________________ **SID#:**________________________

8. (15 pts.) Learning

Consider the problem of deciding whether or not to go on a picnic in England, based on various attributes of the day in question. Here is a set of examples, classified according to whether it was or was not a good idea to go on the picnic. Next to them is one possible decision tree for this problem.

Example	*Rainy*	*Sunny*	*Warm*	*Summer*	*Sunday*	*Picnic*
X_1	T	F	F	F	F	*false*
X_2	F	T	F	F	T	*true*
X_3	F	T	T	T	T	*false*
X_4	F	T	T	F	T	*false*
X_5	T	F	F	F	T	*false*
X_6	F	T	F	F	T	*false*



(a) (2) Circle the false positives for this tree. X_1 X_2 X_3 X_4 X_5 X_6 none

(b) (2) Circle the false negatives for this tree. X_1 X_2 X_3 X_4 X_5 X_6 none

(c) (2) Suppose you could turn one leaf into a new attribute test; which would it be?

(i) $Rainy = T$ (ii) $Sunny = T$ (iii) $Sunny = F$

(d) (3) Which new attribute test provides the highest information gain? [Hint: $\log_2 \frac{1}{3} \approx -1.6$]

(i) $Warm$ (ii) $Summer$ (iii) $Sunday$

(e) (2) *True/False*: The function represented by the decision tree shown above can be represented by a single-layer perceptron using a threshold activation function.

(f) (1) *True/False*: The function represented by the decision tree shown above can be represented by a multi-layer perceptron using a threshold activation function.

(g) (2) *True/False*: The function represented by the decision tree shown above can be represented by a naive Bayes probability model that picks the most likely class.

(h) (1) *True/False*: There is no consistent decision tree for this training set.

9. (10 pts.) Robotic Path Planning

Consider the problem of configuration space path planning in d dimensions using a cell decomposition where each cell is a hypercube. Two cells are adjacent if they share (part of) a face. Here are three possible algorithms, all of which run A* with a straight-line distance heuristic and generate as successors only those cells in the current decomposition that are adjacent to the given cell.

- Algorithm (i) uses a successor function that returns only pure free-space cells.
- Algorithm (ii) uses a successor function that may return mixed free-space/obstacle cells (but no pure obstacle cells).
- Algorithm (iii) is the same as (ii) except that, if the node it selects for expansion corresponds to a mixed cell, it subdivides the cell into hypercubes whose sides are exactly half the size of the original, puts the new cells back in the queue, and repeats the selection step. (For now, assume exact arithmetic.)

(a) (3) Which algorithms are sound (i.e., return only correct solutions)?
(i) (ii) (iii) none

(b) (3) Which algorithms are sound and complete (i.e., return a correct solution exactly when one exists)?
(i) (ii) (iii) none

(c) (1) *True/False*: There are problems on which (i) succeeds but (iii) fails to terminate.

(d) (3) Briefly explain how to make algorithm (iii) ϵ-complete, meaning that it always finds a path if one exists that does not come within ϵ of an obstacle.