CS-171, Intro to A.I. — Final Exam — Spring Quarter, 2011

1. (1 pt) NAME AND EMAIL ADDRESS: ______________________________

YOUR ID: ________ ID TO RIGHT:________ ROW:_____ NO. FROM RIGHT:_____

The exam will begin on the next page. Please, do not turn the page until told.

When you are told to begin the exam, please check first to make sure that you have all eight pages, as numbered 1-8 in the bottom-left corner of each page.

The exam is closed-notes, closed-book. No calculators, cell phones, electronics.

Please clear your desk entirely, except for pen, pencil, eraser, an optional blank piece of paper (for optional scratch pad use), and an optional water bottle.

This page summarizes the points available for each question so you can plan your time.

1. (1 pt) Name and email address.

2. (25 pts total, -1 pt each wrong answer, but not negative) Concepts.

3. (5 points total, 1 pt each) Resolution proof.

4. (5 pts total, -1 pt each wrong answer, but not negative) Search Properties.

5. (5 pts total, -1 for each error, but not negative) Alpha-Beta Pruning.

6. (14 pts total, 2 pt each) FOPL and English correspondences.

7. (13 pts total, 1 pt each) Constraint Satisfaction Problems.

8. (12 pts total) Bayesian Networks.

9. (16 pts total, 2 pt each) Machine Learning.

10. (4 pts total, 1 pt each) Linear classifier (perceptron) learning.

2. (25 pts total, -1 pt each wrong answer, but not negative) Concepts.

For each of the following terms on the left, write in the letter corresponding to the best answer or the correct definition on the right. The first one is done for you as an example.

2.a. Task Environment Concepts. A task environment is defined as a set of four things, with acronym PEAS. Fill in the blanks with the names of the PEAS components.

Performance measure **E**nvironment **A**ctuators **S**ensors

2.b. Agent Concepts.

A	Agent	A	Perceives environment by sensors, acts by actuators
J	Percept	B	Maps any given percept sequence to an action
G	Percept Sequence	C	Agent that acts to maximize its expected performance measure
C	Rational Agent	D	Sensors give the complete state of environment at each time
F	Deterministic Environment	E	Evaluates any given sequence of environment states for utility
H	Dynamic Environment	F	Next state of environment is fixed by current state and action
E	Performance Measure	G	Complete history of everything agent has perceived
B	Agent Function	H	Environment can change while agent is deliberating
I	Abstraction	I	Process of removing detail from a representation
D	Fully Observable	J	Agent's perceptual inputs at any given instant

2.c. Path-Finding Search Concepts.

[During the test: Assume a finite branching factor.]

A	State Space	A	All states reachable from the initial state by a sequence of actions
E	Frontier	B	Guaranteed to find a solution if one is accessible
J	Uninformed Search	C	Never over-estimates cost of cheapest path to a goal state
G	Informed Search	D	Maximum number of successors of any node
I	Optimal Search	E	Set of all leaf nodes available for expansion at any given time
B	Complete Search	F	Estimates cost of cheapest path from current state to goal state
K	Expand a state	G	Uses problem-specific knowledge beyond problem definition
D	Branching Factor	H	Tries to minimize the total estimated solution cost
F	Heuristic Function	I	Guaranteed to find lowest cost among all solutions
H	A* Search	J	Uses no additional information beyond problem definition
M	Greedy Best-first Search	K	Apply each legal action to state, generating a new set of states
L	Consistent Heuristic	L	For n' a successor of n from action a, $h(n) \leq cost(n, a, n') + h(n')$
C	Admissible Heuristic	M	Tries to expand the node believed to be closest to the goal

2.d. Adversarial (Game) Search Concepts.

A	Game Tree	A	Tree where nodes are game states and edges are game moves
H	Cut-off Test	B	In all game instances, total pay-off summed over all players is the same
E	Alpha-Beta Pruning	C	Approximates the value of a game state (i.e., of a game position)
G	Weighted Linear Function	D	Function that specifies a player's move in every possible game state
B	Zero-sum Game	E	Returns same move as MiniMax, but may prune more branches
F	MiniMax Algorithm	F	Optimal strategy for 2-player zero-sum games of perfect information, but impractical given limited time to make each move
D	Game Strategy	G	Vector dot product of a weight vector and a state feature vector
C	Heuristic Evaluation Function	H	Function that decides when to stop exploring this search branch

2.e. Constraint Satisfaction Problem (CSP) Concepts.

A	Solution to a CSP	A	A complete and consistent assignment
D	Complete Assignment	B	Specifies an allowable combination of variable values
G	Constraint Graph	C	Associates values with some or all variables
I	Arc Consistency	D	Every variable is associated with a value
J	Forward Checking	E	The values assigned to variables do not violate any constraints
C	Assignment	F	Set of allowed values for some variable
H	Node Consistency	G	Nodes correspond to variables, links connect variables that participate in a constraint
F	Domain	H	All values in a variable's domain satisfy its unary constraints
B	Constraint	I	All values in a variable's domain satisfy its binary constraints
E	Consistent Assignment	J	When variable X is assigned, delete any value of other variables that is inconsistent with the assigned value of X

2.f. Logic Concepts.

A .	Logic	A	Formal symbol system for representation and inference
E	Valid	B	Specifies all the sentences that are well formed
I	Complete	C	Defines the truth of each sentence in each possible world
C	Semantics	D	The idea that a sentence follows logically from other sentences
G	Conjunctive Normal Form	E	True in every possible world
H	Sound	F	True in at least one possible world
F	Satisfiable	G	A sentence expressed as a conjunction of clauses (disjuncts)
B	Syntax	H	Inference system derives only entailed sentences
D	Entailment	I	Inference system can derive any sentence that is entailed

2.f. Probability concepts and formulae.

A .	Probability Theory	A	Assigns each sentence a degree of belief ranging from 0 to 1
H	Conditional independence	B	Degree of belief accorded without any other information
G	Independence	C	Degree of belief accorded after some evidence is obtained
J	Product rule (chain rule)	D	Gives probability of all combinations of values of all variables
C	Conditional probability	E	Takes values from its domain with specified probabilities
B	Unconditional probability	F	A possible world is represented by variable/value pairs
F	Factored representation	G	$P(a \wedge b) = P(a)\, P(b)$
E	Random variable	H	$P(a \wedge b \mid c) = P(a \mid c)\, P(b \mid c)$
I	Bayes' rule	I	$P(a \mid b) = P(b \mid a)\, P(a) / P(b)$
D	Joint probability distribution	J	$P(a \wedge b \wedge c) = P(a \mid b \wedge c)\, P(b \mid c)\, P(c)$

2.g. Machine Learning concepts.

A .	Learning	A	Improves performance of future tasks after observing the world
H	Regression	B	Fixed set, list, or vector of features/attributes paired with a value
I	Decision Boundary	C	Agent learns patterns in the input with no explicit feedback
K	Overfitting	D	Agent observes input-output pairs & learns to map input to output
J	Cross-validation	E	Example input-output pairs, from which to discover a hypothesis
C	Unsupervised Learning	F	Examples distinct from training set, used to estimate accuracy
B	Factored Representation	G	Supervised learning with a discrete set of possible output values
D	Supervised Learning	H	Supervised learning with numeric output values
F	Test Set	I	Surface in a high-dimensional space that separates the classes
E	Training Set	J	Randomly split the data into a training set and a test set
G	Classification	K	Choose an over-complex model based on irrelevant data patterns

3. (5 points total, 1 pt each) Resolution Proof

Complete the resolution proof below that Jill is Sue's niece. For ease in naming the statements in the KB, they are labeled KB1, KB2, KB3, KB4, etc. Write each unifier as a list of variable/value substitutions, i.e., { var1/val1, var2/val2, …}.

KB1: Daughter(Jill, Mary)

KB2: Sister(Mary, Sue)

KB3: (¬ Daughter(x, y) ∨ ¬ Sister(y, z) ∨ Niece(x, z))

Goal: Niece(Jill, Sue)

(Note: in general, you would need another axiom, Sister(x, y) ⇒ Sister(y, x) *, to state that Sister is symmetric; the problem here has been simplified to avoid that need.)*

The first one is done for you as an example.

3a. The negated goal is KB4: ¬ Niece(Jill, Sue) .

3b. The most general unifier of KB3 and KB4 is: { x/Jill, z/Sue } .

3c. The result of resolving KB3 and KB4 is KB5: ¬ Daughter(Jill, y) ∨ ¬ Sister(y, Sue).

3d. The most general unifier of KB2 and KB5 is: { y/Mary } .

3e. The result of resolving KB2 and KB5 is KB6: ¬ Daughter(Jill, Mary) .

3f. The result of resolving KB1 and KB6 is KB7: {} (or false, or contradiction, etc.).

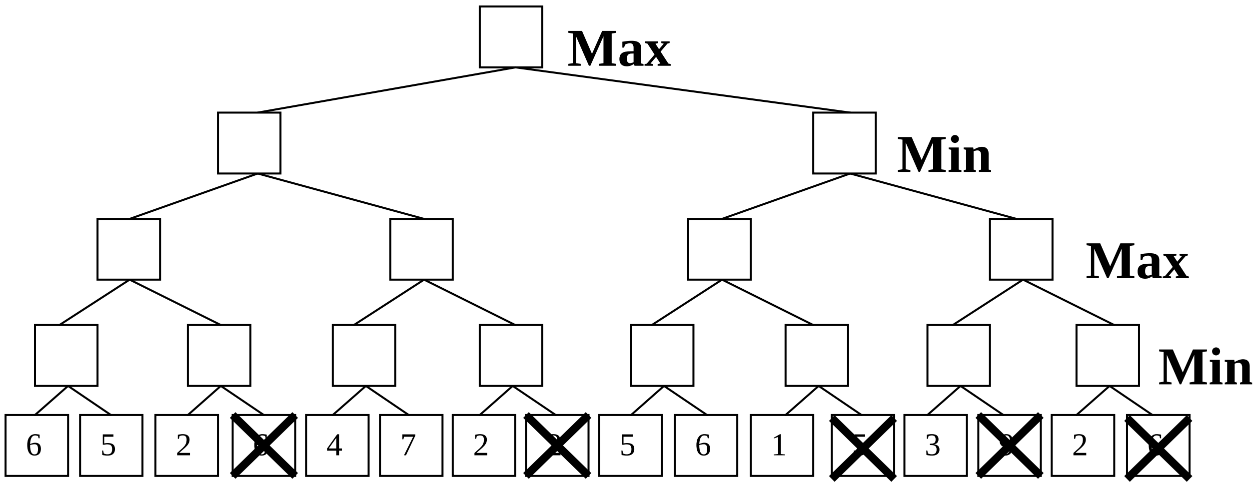
4. (5 pts total, -1 pt each wrong answer, but not negative) Search Properties.

Fill in the values of the four evaluation criteria for each search strategy shown. Assume a tree search where b is the finite branching factor; d is the depth to the shallowest goal node; m is the maximum depth of the search tree; C* is the cost of the optimal solution; step costs are identical and equal to some positive ε; and in Bidirectional search both directions use breadth-first search.

Note that these conditions satisfy all of the footnotes of Fig. 3.21 in your book.

Criterion	Complete?	Time complexity	Space complexity	Optimal?
Breadth-First	Yes	O(b^d)	O(b^d)	Yes
Uniform-Cost	Yes	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	O(b^(1+floor(C*/ε))) O(b^(d+1)) also OK	Yes
Depth-First	No	O(b^m)	O(bm)	No
Iterative Deepening	Yes	O(b^d)	O(bd)	Yes
Bidirectional (if applicable)	Yes	O(b^(d/2))	O(b^(d/2))	Yes

5. (5 pts total, -1 for each error, but not negative) Alpha-Beta Pruning.
Cross out each leaf node that will not be examined because it is pruned by alpha-beta pruning. Traverse the tree left-to-right.



6. (14 pts total, 2 pt each) FOPL and English correspondences.
Fill in each blank below with Y (= Yes) or N (= No) depending on whether the First Order Predicate Logic expression is logically equivalent to the English sentence.

6a. ___N_____ "All men are mortal."
∀x Man(x) ∧ Mortal(x)

6b. ___Y_____ "Fido has a brother who is a dog."
∃x Brother(x, Fido) ∧ Dog(x)

6c. ___Y_____ "For every quiz, there is a student who scored 100 on it."
∀q∃s Quiz(q) ⇒ [Student(s) ∧ Scored100(s, q)]

6d. ___Y_____ "There was no student who scored 100 on every quiz."
∀s∃q Student(s) ⇒ [Quiz(q) ∧ ¬Scored100(s, q)]

6e. ___N_____ "Every student got 100 on every quiz."
∀s∃q Student(s) ⇒ [Quiz(q) ∧ Got100(s, q)]

6f. ___Y_____ "Everyone has a favorite food."
∀x∃y Person(x) ⇒ [Food(y) ∧ Favorite(y, x)]

6g. ___N_____ "There is someone at UCI who is smart."
∃x [Person(x) ∧ At(x, UCI)] ⇒ Smart(x)

7. (13 pts total, 1 pt each) Constraint Satisfaction Problems.
Label the following statements as T (true) or F (false).

7a. __T____ A **constraint satisfaction problem** (CSP) consists of a set of variables, a set of domains (one for each variable), and a set of constraints that specify allowable combinations of values.

7b. __F____ A **consistent assignment** is one in which every variable is assigned.

7c. __F____ A **complete assignment** is one that does not violate any constraints.

7d. __F____ A **partial assignment** is one that violates only some of the constraints.

7e. __T____ The nodes of a **constraint graph** correspond to variables of the problem, and a link connects any two variables that participate in a constraint.

7f. __T____ A **constraint** consists of a pair <*scope, rel*>, where *scope* is a tuple of variables that participate and *rel* defines the values those variables can take on.

7g. __T____ Performing **constraint propagation** involves using the constraints to reduce the number of legal values for a variable, which in turn can reduce the legal values for another variable, and so on.

7h. __T____ A variable in a CSP is **arc-consistent** iff, for each value in its domain and each of its binary constraints, that constraint is satisfied by that domain value together with some value in the domain of the other variable in that constraint.

7i. __F____ Constraint satisfaction problems are **semi-decidable** because they may never terminate if the problem has no legal solution.

7j. __T____ The **minimum-remaining-values** (MRV) heuristic chooses the variable with the fewest remaining legal values to assign next.

7k. __F____ The **degree heuristic** is used to set the temperature in methods for solving CSPs based on Simulated Annealing.

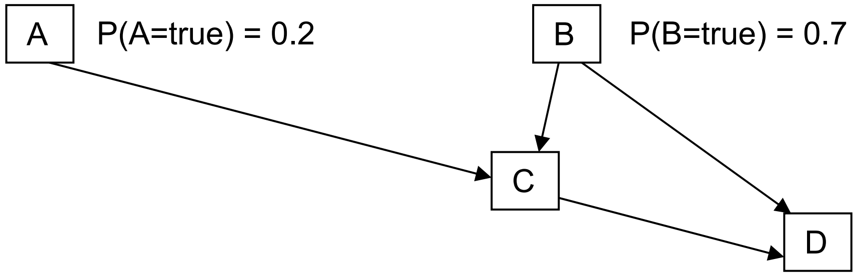
7l. __T____ The **least-constraining-value** heuristic prefers the value that rules out the fewest choices for the neighboring variables in the constraint graph.

7m. __T____ The **min-conflicts** heuristic for local search prefers the value that results in the minimum number of conflicts with other variables.

7n. __F____ The **min-conflicts** heuristic is rarely used because it is only effective when the constraint graph is a tree.

8. (12 pts total) Bayesian Networks.

Consider the following Bayesian Network. Variables A-D are Boolean:



A	B	P(C=true | A, B)
false	false	0.1
false	true	0.5
true	false	0.4
true	true	0.9

B	C	P(D=true | B, C)
false	false	0.8
false	true	0.6
true	false	0.3
true	true	0.1

8.a. (2 pt) Use the chain rule to factor the full joint probability distribution over these variables into a product of conditional probabilities, ignoring conditional independence from the figure. Factor out the conditional probability of D first, C second, etc.

P(A, B, C, D) = <u>P(D | C, B, A) P(C | B, A) P(B | A) P(A)</u>

8.b. (6 pts) Use the structure of the network to eliminate irrelevant variables from 8.a based on conditional independence, giving the minimum equivalent expression.

P(A, B, C, D) = <u>P(D | C, B) P(C | B, A) P(B) P(A)</u>

8.c. (4 pt) Substitute probabilities from the network into your equation 8.b to answer the query: What is the probability that all four of these Boolean variables are false?

$P(\neg a, \neg b, \neg c, \neg d) = P(\neg d \mid \neg c, \neg b)\, P(\neg c \mid \neg b, \neg a)\, P(\neg b)\, P(\neg a) = 0.2 * 0.9 * 0.3 * 0.8$

= <u>0.0432</u>

9. (16 pts total, 2 pt each) Machine Learning.
Label the statements T (true) or F (false).

9a. __F__ A linear classifier (perceptron) can learn and represent any Boolean function.

9b. __T___ A decision tree can learn and represent any Boolean function.

9c. ___F___ A Naïve Bayes classifier can learn and represent only axis-parallel classes.

9d. ___F___ "Naive Bayes" is called "naive" because only naive people ever use it.

9e. ___T___ Overfitting is a general phenomenon that occurs with all types of learners.

9f. ___F___ The information gain from an attribute A is how much classifier accuracy improves when attribute A is added to the example feature vectors in the training set.

9g. ___T___ An agent is learning if it improves its performance on future tasks after making observations about the world.

9h. ___F___ Cross-validation is a way to improve the accuracy of a learned hypothesis by reducing over-fitting using Ockham's razor.

10. (4 pts total, 1 pt each) Linear classifier (perceptron) learning.
Label the statements Y (yes) or N (no).

10a. __N___ Suppose that you are given two weight vectors for a perceptron. Both vectors, w1 and w2, correctly recognize a particular class of examples. Does the vector w3 = w1 − w2 ALWAYS correctly recognize that same class?

10b. __Y___ Does the vector w4 = w1 + w2 ALWAYS correctly recognize that same class?

10c. __Y___ Does the vector w5 = cw1 where c = 42 ALWAYS correctly recognize that same class?

10e. __N___ Suppose that you are given two examples of the same class A, x1 and x2, where x1 ≠ x2. Suppose the example x3 = 0.5x1 + 0.5x2 is of a different class B. Is there ANY perceptron that can classify x1 and x2 into class A and x3 into class B?