For the Final Exam, “Perfect” gives the percentage of students who received full credit, “Partial” gives the percentage who received partial credit, and “Zero” gives the percentage who received zero credit.

(Due to rounding, etc., values below may be only approximate estimates.)

Problem 1
Perfect: ~76% (~148 students), Partial: ~18% (~35 students), Zero: ~7% (~13 students)
A common mistake was to select m=3 and so merge the training and testing error. Another common mistake was to calculate MSE as $(1/m)\ (\Sigma_j\ (Yj - \bar{Y}j)\ )^2$ instead of the correct $(1/m)\ \Sigma_j\ (Yj - \bar{Y}j)^2$.

Problem 2
Perfect: ~95% (~187 students), Partial: ~5% (~9 students), Zero: ~0% (~0 students)

Problem 3.a
Perfect: ~40% (~78 students), Partial: ~35% (~69 students), Zero: ~25% (~49 students)
A common mistake was to switch cluster #1 and cluster #2. Pay attention to their evolution over time. Another common mistake was not to understand K-means clustering at all, leading to lost points.

Problem 3.b
Perfect: ~34% (~67 students), Partial: ~58% (~114 students), Zero: ~8% (~15 students)
A common mistake was not to understand what the cluster dendogram (= tree) was supposed to be.

Problem 4
Perfect: ~20% (~40 students), Partial: ~80% (~156 students), Zero: ~0% (~0 students)
A common mistake on 4.d.1 was to mention only T1 instead of T1, T3, and T4.

Problem 5
Perfect: ~94% (~184 students), Partial: ~6% (~11 students), Zero: ~0.5% (~1 students)
A common mistake was an incorrect or trivial resolution step. Incorrect or trivial examples include:

Resolve	(A C)	with	((¬ A) (¬ B))	to produce:	(A (¬ C))	[wrong]
Resolve	(A D)	with	((¬ C) (¬ D))	to produce:	(A)	[wrong]
Resolve	(A (¬ C))	with	((¬ A) C)	to produce:	(C (¬ C))	[OK, but trivial]
Resolve	(A B)	with	((¬ B) (¬ C))	to produce:	(¬ C)	[wrong]

Problem 6
Perfect: ~41% (~81 students), Partial: ~59% (~115 students), Zero: ~0% (~0 students)
Common mistakes were:
For 6.c, writing “∃x ∃y Person(x) ∧ Game(y) => Plays(x, y)”

> This sentence is vacuously true if there is anything in the world that is not a person (x) or is not a game (y), because that x or that y will make the antecedent false and the implication true.

For 6.e and 6.f, using the wrong parentheses. Using the wrong parentheses always causes trouble.

Problem 7
Perfect: ~83% (~163 students), Partial: ~17% (~33 students), Zero: ~0% (~0 students)
Common mistakes were:
Problem 7.a: Omit P(H).
Problem 7.b: Draw the arrow in the opposite (wrong) direction.
Problem 7.c: Omit P(B=t) & P(E=f); or write too simply, P(B) & P(E), without giving =true or =false.

Problem 8
Perfect: ~54% (~106 students), Partial: ~39% (~77 students), Zero: ~7% (~13 students)
A common mistake was on depth-limited search, which became confused with other searches.

Another common mistake was for students to have forgotten entirely what they learned in the first part of the course, and not to have studied the class material again so as to refresh their memories for the Final Exam. Please consult Figure 3.21, page 91, of your textbook.

CS-171, Intro to A.I. — Final Exam — Fall Quarter, 2015

YOUR NAME: ____

YOUR ID: __________ ID TO RIGHT: __________ ROW: _________ SEAT: ________

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

When you are told to begin the exam, please check first to make sure that you have all eleven pages, as numbered 1-11 in the bottom-right corner of each page. We wish to avoid copy problems. We will supply a new exam for any copy problems.

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

<u>Please turn off all cell phones now. No electronics are allowed at any point of the exam.</u>

Please clear your desk entirely, except for pen, pencil, eraser, a blank piece of paper (for scratch pad use), and an optional water bottle. Please write your name and ID# on the blank piece of paper and <u>turn it in with your exam</u>.

<u>This page summarizes the points for each question, so you can plan your time.</u>

1. (15 pts total, 3 pts each) Linear Regression.

2. (4 pts total, 1 pt each) Task Environment.

3. (18 pts total) Clustering.

4. (16 pts total) Constraint Satisfaction Problems and Job Shop Scheduling.

5. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Resolution Theorem Proving.

6. (15 pts total, 3 pts each) English to FOL Conversion.

7. (12 pts total, 4 pts each) Bayesian Networks.

8. (10 pts total, -1 pt each error, but not negative) Search Properties.

<u>The Exam is printed on both sides to save trees! Work both sides of each page!</u>

1. (15 pts total, 3 pts each) Linear Regression.
Suppose we have following dataset for a linear regression problem.

See sections 18.6.1-2, pages 718-723, in your textbook.

Training data

Example	**Feature *X1***	**Feature *X2***	**Target *Y***
Example #1	3	5	18
Example #2	7	2	28

Testing data

Example	**Feature *X1***	**Feature *X2***	**Target *Y***
Example #3	4	1	18

There are two features in this problem and so we have three parameters (weights) in this regression problem. Parameter $\Theta 0$ is the constant weight. Parameters $\Theta 1$ and $\Theta 2$ are the weights of features $X1$ and $X2$ respectively. Recall that the resulting linear regression model is $\bar{Y} = \Theta 0 + \Theta 1 * X1 + \Theta 2 * X2$.

For this problem, assume that you are given ***Weights*** $= [\Theta 0 = 1, \Theta 1 = 3, \Theta 2 = 2]$.

1.a (9 pts total, 3 pts each) Given ***Weights*** above, what is the predicted value for the three examples above?

<u>SHOW YOUR WORK.</u> First display the formula above for Ȳ after substituting numbers for Θi and Xi (this step is worth most or all of the points). Then simplify to obtain a numerical answer. The numbers have been chosen to be so simple that you do not need a calculator.

Predicted value for Example #1:

Ȳ1 = 1 + 3 * 3 + 2 * 5 = 1 + 9 + 10 = 20

Predicted value for Example #2:

Ȳ2 = 1 + 3 * 7 + 2 * 2 = 1 + 21 + 4 = 26

Predicted value for Example #3:

Ȳ3 = 1 + 3 * 4 + 2 * 1 = 1 + 12 + 2 = 15

1.b (6 pts total, 3 pts each) Calculate the value of Mean Squared Error (MSE) for training and testing data separately. Recall that the MSE formula is $MSE = (1/m)\ \Sigma_j\ (Yj - \bar{Y}j)^2$, where m is the number of examples.

<u>SHOW YOUR WORK.</u> First display the formula above for MSE after expanding the summation and substituting numbers for m, Yj, and <u>Ȳj</u> (this step is worth most or all of the points). Then simplify to obtain a numerical answer. The numbers have been chosen to be so simple that you do not need a calculator. Your answer to **1.b** is correct if it is correct relative to your answer to **1.a**, even if your answer to **1.a** was wrong.

Mean Squared Training Error:

$MSE = (1/2)\ [\ (18 - 20)^2 + (28 - 26)^2\ ] = (1/2)\ [\ 2^2 + 2^2] = (1/2)\ [\ 4 + 4\ ] = 4$

Mean Squared Testing Error:

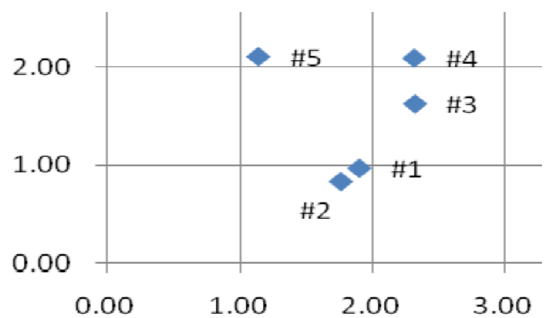
$MSE = (1/1)\ [\ (18 - 15)^2\ ] = (1/1)\ [\ 3^2\ ] = (1/1)\ [\ 9] = 9$

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

2. (4 pts total, 1 pt each) Task Environment. Your book defines a task environment as a set of four things, with acronym PEAS. Fill in the blanks with the names of the PEAS components.

Performance (measure) Environment ______ Actuators ______ Sensors ______

3. (18 pts total) Clustering. In this problem you will cluster the following data points, first by K-Means Clustering (3.a) and then by Hierarchical Agglomerative Clustering (3.b). Use Euclidean distance.



Data Point	x	y
Example #1	1.90	0.97
Example #2	1.76	0.84
Example #3	2.32	1.63
Example #4	2.31	2.09
Example #5	1.14	2.11

See class lecture notes on Clustering.

Distances	#1	#2	#3	#4	#5
#1	-	0.191	0.782	1.193	1.370
#2	-	-	0.968	1.366	1.413
#3	-	-	-	0.460	1.274
#4	-	-	-	-	1.170

3.a. (8 pts total, 2 pts each) K-Means Clustering. Assume the number of clusters is fixed at TWO (K = 2). Assume that the initial random assignment of cluster means (= cluster centers) assigns ClusterMean1 = Example #1 and ClusterMean2 = Example #2. Recall that each example is assigned to its nearest cluster mean (= cluster center).

3.a.1 (4 pts total, 2 pts each) <u>Write the example #s that initially are assigned to those clusters,</u> assuming initial cluster means as above. Write your answer as a list of examples in the #i format as given above, i.e., as: #i, #j, #k, ...

(Initial ClusterMean1 = Example #1)
Initial Cluster1 Members = ______ #1, #3, #4, #5 ______

(Initial ClusterMean2 = Example #2)
Initial Cluster2 Members = ______ #2 ______

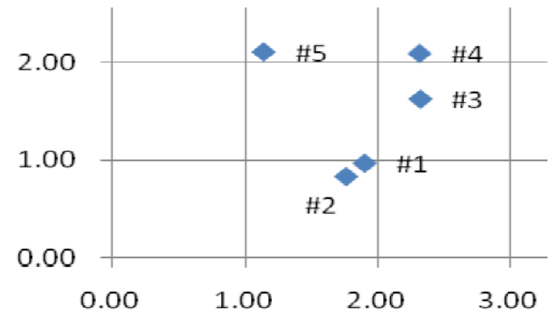
3.a.2 (4 pts total, 2 pts each) Hand-simulate the K-means algorithm until it converges (this happens on the next iteration). The numbers have been chosen to be so simple that you do not need a calculator.

<u>Write the final example #s found in those clusters when K-means converges.</u> Write your answer as a list of examples in the #i format as given above, i.e., as: #i, #j, #k, ...

Final Cluster1 Members = ______ #3. #4, #5 ______

Final Cluster2 Members = ______ #1, #2 ______

3.b. (10 pts total) Hierarchical Agglomerative Clustering. Problem 3.b uses the same data set as problem 3.a.

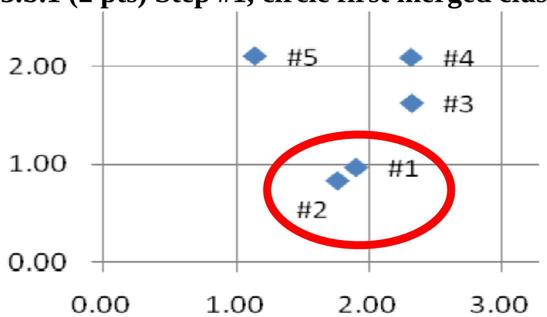


Data Point	x	y
Example #1	1.90	0.97
Example #2	1.76	0.84
Example #3	2.32	1.63
Example #4	2.31	2.09
Example #5	1.14	2.11

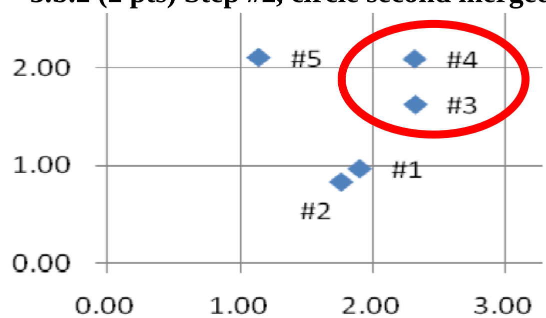
Distances	#1	#2	#3	#4	#5
#1	-	0.191	0.782	1.193	1.370
#2	-	-	0.968	1.366	1.413
#3	-	-	-	0.460	1.274
#4	-	-	-	-	1.170

3.b.1-4 (2 pts each) Hand-simulate hierarchical agglomerative clustering until it converges (this happens after four steps). **Use Min Distance for Cluster Distance,** i.e., distance between two clusters is the minimum distance from any point in one cluster to any point in the other cluster. Initially, each point is one cluster. Then, at each step, the two closest clusters are merged to create a new cluster. **At each step, circle the points in the new merged cluster formed at that step.**

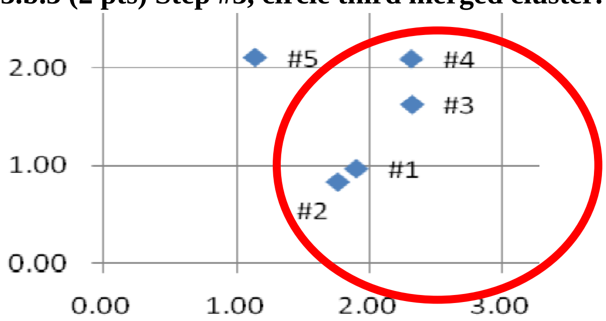
3.b.1 (2 pts) Step #1, circle first merged cluster.



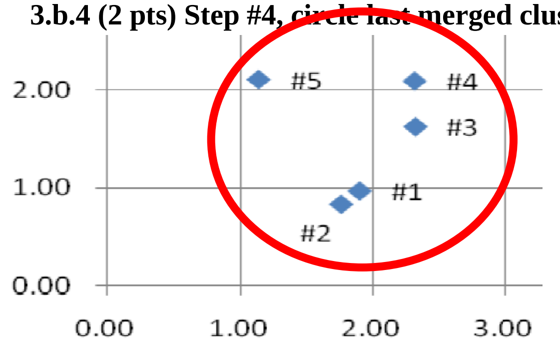
3.b.2 (2 pts) Step #2, circle second merged cluster.



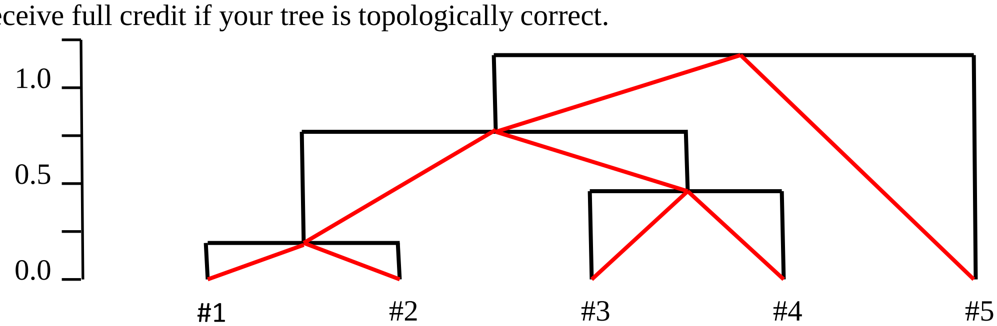
3.b.3 (2 pts) Step #3, circle third merged cluster.



3.b.4 (2 pts) Step #4, circle last merged cluster.



3.b.5 (2 pts) Draw the cluster dendogram that results. The x-axis gives the example #s. The y-axis gives the distance between the merged clusters at which the new cluster was formed. It is OK to give only very approximate y values, i.e. you will receive full credit if your tree is topologically correct.



The red and black lines show two alternate ways to draw the tree. Either way, red tree or black tree, is correct. The black tree is far more common, and is usually much easier to interpret.

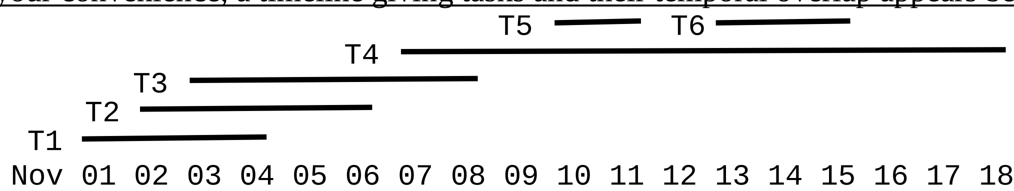
**** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ****

4. (16 pts total) Constraint Satisfaction Problems and Job Shop Scheduling. You are a job shop scheduling robot that must assign workers to tasks in order to meet scheduling constraints. You have a list of tasks that must be done, a list of workers who can do each task, and a list of times at which each task must be done. (In general, such planning problems are more complicated than our constraint satisfaction problems because their constraints can change as time passes, but this problem has been simplified to match the material you studied.)

Your list of 6 tasks and when they must be done is:

- Task T1: must be done from Nov 1 to Nov 4. It will occupy one worker for all 4 days.
- Task T2: must be done from Nov 2 to Nov 6. It will occupy one worker for all 5 days.
- Task T3: must be done from Nov 3 to Nov 8. It will occupy one worker for all 6 days.
- Task T4: must be done from Nov 7 to Nov 18. It will occupy one worker for all 12 days.
- Task T5: must be done from Nov 10 to Nov 11. It will occupy one worker for all 2 days.
- Task T6: must be done from Nov 13 to Nov 15. It will occupy one worker for all 3 days.

For your convenience, a timeline giving tasks and their temporal overlap appears below:



Your list of 3 workers and the tasks they can do is:

- Worker W1: can do tasks T3, T4, T5, and T6.
- Worker W2: can do tasks T1, T2, T4, T5, and T6.
- Worker W3: can do tasks T1, T3, T5 and T6.

See sections 6.1-6.4 in your textbook.

Your scheduling constraints are:

- Every task must be started on its assigned start date.
- Every task must continue for the length of time stated above for that task.
- Every task must be assigned to a worker who can do that task.
- Once a worker is assigned to a task that worker must work only on that task until finished.
- Once a worker is assigned to a task that worker cannot be assigned to another task until finished.
- Only one worker may be assigned to each task.
- Therefore, tasks that overlap in time cannot be assigned to the same worker.

You are overwhelmed by the complexity of your assignment. So, you decide to formulate your assignment as a Constraint Satisfaction Problem. You create **one variable for each task**, and so you have six variables, T1-T6. The **domain of each variable is the set of workers who can do that task**. The **constraints are that every pair of tasks that overlap in time must have a different worker** for each such task (similar to the AllDiff constraint in Sudoku problems). Solving the resulting Constraint Satisfaction Problem yields your job schedule.

4.a (6 pts total, 1 pt each). Enumerate the domains of each variable. For each task variable T1-T6 named below, list the domain values of the workers W1-W3 who can do that task.

4.a.1 (1 pt) Task T1: domain = W2, W3

4.a.2 (1 pt) Task T2: domain = W2

4.a.3 (1 pt) Task T3: domain = W1, W3

4.a.4 (1 pt) Task T4: domain = W1, W2

4.a.5 (1 pt) Task T5: domain = W1, W2, W3

4.a.6 (1 pt) Task T6: domain = W1, W2, W3

4.b (6 pts total, 1 pt each). Enumerate the constraints among overlapping variables. For each scope (= pair of task variables T1-T6 that overlap in the timeline listed above), **list the consistent pairs of variable values,** i.e., those pairs that are allowed by the constraints. One constraint is done for you as an example.

4.b.1 Scope = (T4, T6). Consistent pairs = (W1, W2), (W1, W3), (W2, W1), (W2, W3)

4.b.2 (1 pt) Scope = (T1, T2). Consistent pairs = (W3, W2)

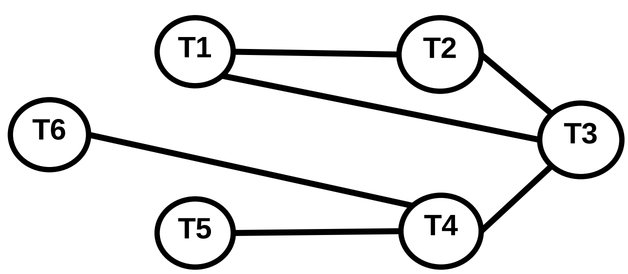
4.b.3 (1 pt) Scope = (T1, T3). Consistent pairs = (W2, W1), (W2, W3), (W3, W1)

4.b.4 (1 pt) Scope = (T2, T3). Consistent pairs = (W2, W1), (W2, W3)

4.b.5 (1 pt) Scope = (T3, T4). Consistent pairs = (W1, W2) , (W3, W1), (W3, W2)

4.b.6 (1 pt) Scope = (T4, T5). Consistent pairs = (W1, W2), (W1, W3), (W2, W1), (W2, W3)

4.c (1 pt total) Draw the constraint graph for variables T1-T6.



4.d (4 pts total, 2 pts each) Variable ordering during Backtracking Search. Assume that T2 is already assigned to its only domain value. Assume that all other variables are unassigned and have the domain values listed in (4.a) above. Assume that no constraint propagation has been done.

4.d.1 (2 pts) What variables might be assigned by the Minimum Remaining Values (MRV) heuristic:

T1, T3, T4,

4.d.2 (2 pts) What variables might be assigned by the Degree Heuristic (DH) heuristic:

T4,

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

5. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Resolution Theorem Proving. (With apologies to Dr. Seuss.)

Amy, Betty, Cindy, and Diane went out to lunch at a seafood restaurant. Each ordered one fish. Each fish was either a red fish or a blue fish.

See R&N Section 7.5.2.

Their waiter reported:

"Amy and Betty had fish of different colors; I don't remember who had what.
Betty and Cindy had fish of different colors; I don't remember who had what.
Cindy and Diane had fish of different colors; I don't remember who had what.
Amy, Cindy, and Diane had exactly two red fish among them; I don't remember who had what."

You translate these facts into Propositional Logic (in infix form) as:
/* Ontology: Symbol A/B/C/D means that Amy/Betty/Cindy/Diane had a red fish. */
(A <=> (¬ B))   (B <=> (¬ C))   (C <=> (¬ D))
((A ∧ C ∧ (¬ D)) ∨ (A ∧ (¬ C) ∧ D) ∨ ((¬ A) ∧ C ∧ D))

Betty's daughter asked, "Is it true that my mother had a blue fish?"
You translate this query into Propositional Logic as "(¬ B)" and form the negated goal as "(B)".

Your resulting knowledge base (KB) plus the negated goal (in CNF clausal form) is:
(A B)   (A C)   (A D)   (B C)   (C D)
((¬ A) (¬ B))   ((¬ B) (¬ C))   ((¬ C) (¬ D))
(B)

Write a resolution proof that Betty had a blue fish.
For each step of the proof, fill in the first two blanks with CNF sentences from KB that will resolve to produce the CNF result that you write in the third (resolvent) blank. The resolvent is the result of resolving the first two sentences. Add the resolvent to KB, and repeat. Use as many steps as necessary, ending with the empty clause. The empty clause indicates a contradiction, and therefore that KB entails the original goal sentence.

The shortest proof that I know of is only four lines long. (A Bonus Point is offered for a shorter proof.) Longer proofs are OK provided they are correct. *Think about it, then find a proof that mirrors how you think. Obviously, the four of them must have had two red fish and two blue fish among them. Obviously, if Amy, Cindy, and Diane had two red fish among them, then Betty must have had a blue fish.*

Resolve __(A C)__ with __((¬ C) (¬ D))__ to produce: __(A (¬ D))__

Resolve __(A (¬ D))__ with __(A D)__ to produce: __(A)__

Resolve __(B)__ with __((¬ A) (¬ B))__ to produce: __(¬ A)__

Resolve __(A)__ with __(¬ A)__ to produce: __( )__

It is OK if you omitted the parentheses.

Twenty-nine bright and clever students found a 3-line proof shorter than I had been able to find (and so won a Bonus Point). One student found both of the 3-line proofs below (and so won two (2) Bonus Points).

The first 3-line proof was:

Resolve __(A C)__ with __((¬ B) (¬ C))__ to produce: __(A (¬ B))__

Resolve __(A (¬ B))__ with __((¬ A) (¬ B))__ to produce: __(¬ B)__

Resolve __(¬ B)__ with __(B)__ to produce: __( )__

The second 3-line proof was:

Resolve ______(A C)__________ with __((¬ A) (¬ B))________________ to produce: __((¬ B) C)___________

Resolve ______((¬ B) C)_____ with __((¬ B) (¬ C))________________ to produce: __(¬ B)________________

Resolve ______(¬ B)__________ with __(B)___________________________ to produce: __( )__________________

6. (15 pts total, 3 pts each) English to FOL Conversion. For each English sentence below, write the FOL sentence that best expresses its intended meaning. Use Person(x) for "x is a person," Game(x) for "x is a game," and Plays(x, y) for "x plays y."

The first one is done for you as an example.

See Section 8.2.6

6.a. "Every person plays every game."

$\forall x\ \forall y\ [\ Person(x) \wedge Game(y)\ ] \Rightarrow Plays(x, y)$

6.b. (3 pts) "Some person plays some game."

$\exists x\ \exists y\ Person(x) \wedge Game(y) \wedge Plays(x, y)$

6.c. (3 pts) "For every person, there is a game that the person plays."

$\forall x\ \exists y\ Person(x) \Rightarrow [\ Game(y) \wedge Plays(x, y)\ ]$

6.d. (3 pts) "For every game, there is a person who plays that game."

$\forall y\ \exists x\ Game(y) \Rightarrow [\ Person(x) \wedge Plays(x, y)\ ]$

6.e. (3 pts) "There is a game that every person plays."

$\exists y\ \forall x\ Game(y) \wedge [\ Person(x) \Rightarrow Plays(x, y)\ ]$

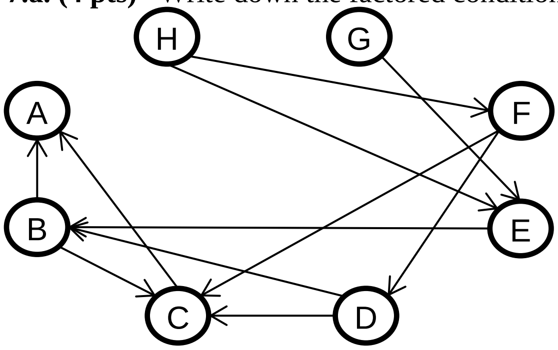
6.f. (3 pts) "There is a person who plays every game."

$\exists x\ \forall y\ Person(x) \wedge [Game(y) \Rightarrow Plays(x, y)\ ]$

****** TURN PAGE OVER AND CONTINUE ON THE OTHER SIDE ******

7. (12 pts total, 4 pts each) BAYESIAN NETWORKS.

7.a. (4 pts) Write down the factored conditional probability expression corresponding to this Bayesian Network:

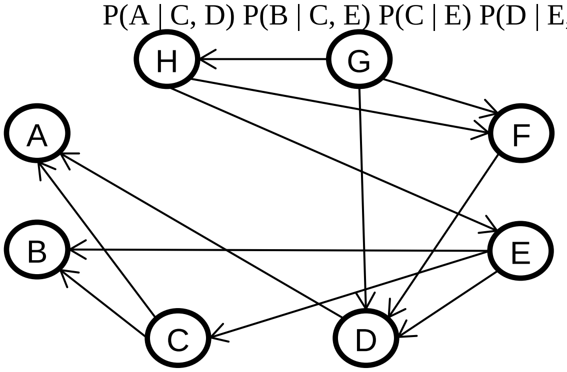


See Sections 14.1-14.5.

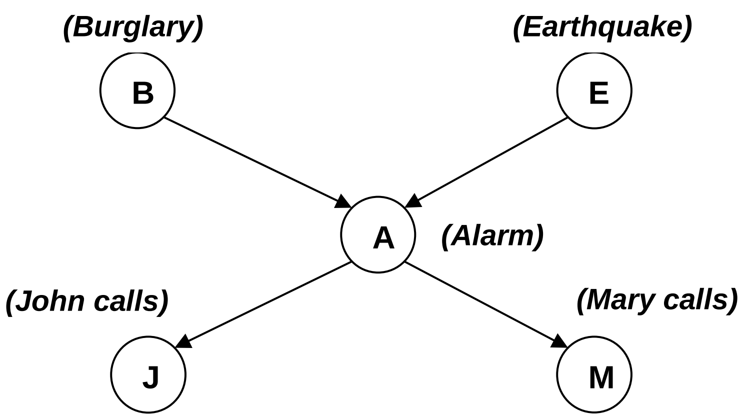
P(A | B, C) P(B | D, E) P(C | B, D, F) P(D | F) P(E | G, H) P(F | H) P(G) P(H)

7.b. (4 pts) Draw the Bayesian Network corresponding to this factored conditional probability expression:

P(A | C, D) P(B | C, E) P(C | E) P(D | E, F, G) P(E | H) P(F | G, H) P(G) P(H | G)



7.c. (4 pts) Shown below is the Bayesian network corresponding to the Burglar Alarm problem, i.e., P(J,M,A,B,E) = P(J | A) P(M | A) P(A | B, E) P(B) P(E). This is Fig. 14.2 in your R&N textbook.



P(B)
.001

P(E)
.002

B	E	P(A)
t	t	.95
t	f	.94
f	t	.29
f	f	.001

A	P(J)
t	.90
f	.05

A	P(M)
t	.70
f	.01

Write an expression that will evaluate to P(J=f ∧ M=t ∧ A=t ∧ B=t ∧ E=f). **Express your answer first as the product of symbolic probabilities, then as a series of numbers (numerical probabilities) separated by multiplication symbols.** You do not need to carry out the multiplication. The first probability of each line is done for you as an example.

P(J=f ∧ M=t ∧ A=t ∧ B=t ∧ E=f)

= P(J=f | A=t) * P(M=t | A=t) * P(A=t | B=t ∧ E=f) * P(B=t) * P(E=f)

= .10 * .70 * .94 * .001 * .998

8. (10 pts total, -1 pt each error, 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 and may be infinite; l is the depth limit; step costs are identical and equal to some positive ε; in bidirectional search both directions use breadth-first search.

Note: These assumptions are the same as in Figure 3.21 of your textbook.

	Complete?	**Time complexity**	**Space complexity**	**Optimal?**
Depth-First	No	O(b^m)	O(bm)	No
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-Limited	No	O(b^l)	O(bl)	No
Iterative Deepening	Yes	O(b^d)	O(bd)	Yes
Bidirectional (if applicable)	Yes	O(b^(d/2))	O(b^(d/2))	Yes

Your answer will be considered correct if it differs from that shown above by no more than ±1, e.g., O(b^d) vs. O(b^(d+1)).

This table is the same as Figure 3.21 R&N.

****** THIS IS THE END OF THE FINAL EXAM ******