CS-171, Intro to A.I. — Final Exam — Winter 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 13 pages, as numbered 1-13 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.</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. (10 pts total, 1 pt each) MACHINE LEARNING.

2. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Resolution Theorem Proving. (With apologies to Dr. Seuss.)

3. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Naïve Bayes Classifier Learning. (With apologies to Dr. Suess.)

4. (10 pts total, 2 pts each) WUMPUS WORLD MODELS.

5. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Decision Tree Classifiers. (With apologies to Dr. Suess.)

6. (10 points total, 2 pts each) CONSTRAINT SATISFACTION PROBLEMS.

7. (10 pts total) BAYESIAN NETWORKS.

8. (10 pts total, 2 pts each) STATE-SPACE SEARCH.

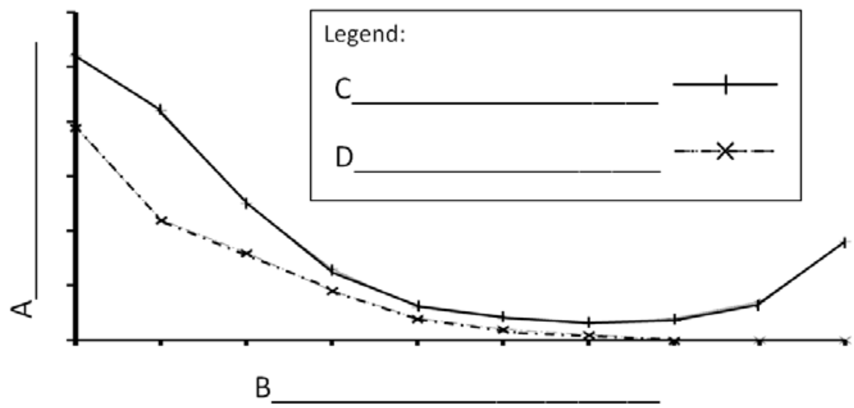
9. (10 pts total, -1 pt for each error, but not negative) GAME (ADVERSARIAL) SEARCH.

10. (10 pts total, 1 pt each) FOPC KNOWLEDGE ENGINEERING IN THE TOY BLOCKS WORLD.

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

1. (10 pts total, 1 pt each) Machine Learning.

1.a. (4 pts total, 1 pt per blank) While studying for the Final Exam, you tried to print out Fig. 18.9 from the text book. That figure explains how error rates change with tree size (or model complexity). However, for some unknown reason, the figure legend and axis labels did not print out correctly.



Fill in the blanks below with A, B, C, or D to indicate where each label below belongs on the graph above.

_________ (Write A, B, C, or D.) Tree size (or Model complexity)

_________ (Write A, B, C, or D.) Training set error

_________ (Write A, B, C, or D.) Validation set error

_________ (Write A, B, C, or D.) Error rate

1.b. (3 pts total, 1 pt each) While working on a traffic sign recognition problem, you implemented four different classifiers. You are now trying to compare their performance. Their misclassification rates are as follows:

	Classifier A	Classifier B	Classifier C	Classifier D
Error rate on training data	25%	5%	10%	20%
Error rate on testing data	30%	20%	15%	25%

1.b.i. (1pt) _________ (Write A, B, C, or D.) Which classifier has the best generalization performance, i.e., most likely would perform the best when applied to unseen data?

1.b.ii. (1pt) _________ (Write A, B, C, or D.) Which classifier is underfitting the most?

1.b.iii. (1pt) _________ (Write A, B, C, or D.) Which classifier is overfitting the most?

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

1.c. (3 pts total; 1pt each). In the figures below, each data point has Class either positive (+) or negative (–). For each of the two-dimensional data sets below, put an "X" in the box of any classifier that can perfectly separate the positive from the negative data points. For example, if a Linear Perceptron can separate the classes perfectly, put an "X" in its box: ☒ Linear Perceptron. In most or all cases, more than one classifier can separate the data set.

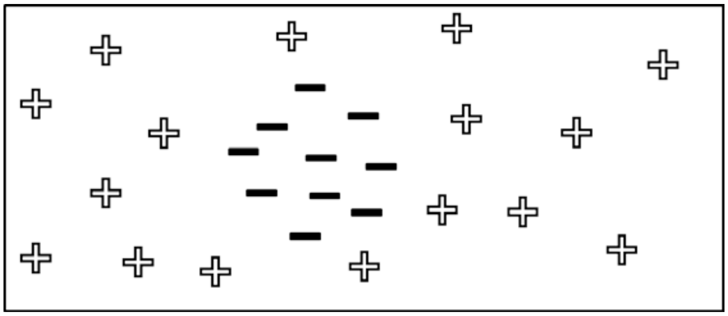
1.c.i. (1 pt)

☐ Linear Perceptron

☐ Nearest Neighbor

☐ Decision Tree

☐ Support Vector Machine



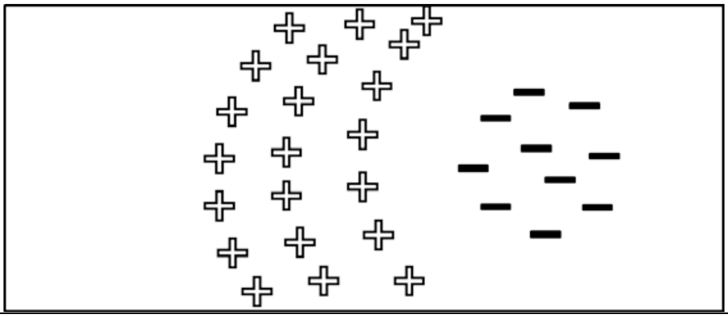
1.c.ii. (1 pt)

☐ Linear Perceptron

☐ Nearest Neighbor

☐ Decision Tree

☐ Support Vector Machine



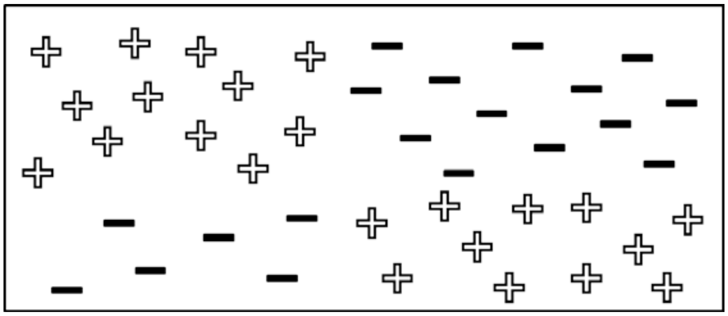
1.c.iii. (1 pt)

☐ Linear Perceptron

☐ Nearest Neighbor

☐ Decision Tree

☐ Support Vector Machine



2. (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. **Among them they had exactly two red fish and two blue fish.**

You translate this fact into Propositional Logic (in prefix form) as:

<u>/* Ontology: Symbol A/B/C/D means that Amy/Betty/Cindy/Diane had a red fish. */</u>

```
(OR  (AND A B (¬ C) (¬ D) )          (AND A (¬ B) C (¬ D) )
     (AND A (¬ B) (¬ C) D)           (AND (¬ A) B C (¬ D) )
     (AND (¬ A) B (¬ C) D)           (AND (¬ A) (¬ B) C D) )
```

Their waiter reported:

"Amy and Diane had the same color fish; I don't remember which color they were.
Amy, Betty, and Cindy had exactly one red fish among them; I don't remember who had what."

You translate these facts into Propositional Logic (in prefix form) as:

```
(<=> A D)
(OR  (AND A (¬ B) (¬ C) )       (AND (¬ A) B (¬ C) )       (AND (¬ A) (¬ B) C) )
```

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)".

<u>Your resulting knowledge base (KB) plus the negated goal (in CNF clausal form) is:</u>

```
(A B C)              ( (¬ A) (¬ B) (¬ C) )
(A B D)              ( (¬ A) (¬ B) (¬ D) )
(A C D)              ( (¬ A) (¬ C) (¬ D) )
(B C D)              ( (¬ B) (¬ C) (¬ D) )
( (¬ A) D)           (A (¬ D) )
(A B C)              ( (¬ A) (¬ B))
( (¬ A) (¬ C))       ( (¬ B) (¬ C))
(B)
```

<u>**Write a resolution proof that Betty had a blue fish.**</u>

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 five lines long. (A Bonus Point is offered for a shorter proof.) Longer proofs are OK provided they are correct. <u>*Think about it, then find a proof that mirrors how you think. Obviously, Amy and Diane must have had red fish, while Betty and Cindy had blue fish.*</u>

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

Resolve ________________ with __________________________ to produce: _____________

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

3. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Naïve Bayes Classifier Learning. (With apologies to Dr. Suess.) You are a robot in the aquarium section of a pet store, and must learn to discriminate Red fish from Blue fish. Unfortunately, your vision sensors are in Black & White, but Red fish have the same gray-scale tone as Blue fish. So, you must learn to discriminate them by body parts. You choose to learn a Naïve Bayes classifier. You are given the following examples:

Example	Fins	Tail	Body	Class
Example #1	Thin	Large	Slim	Red
Example #2	Wide	Small	Slim	Red
Example #3	Wide	Large	Fat	Red
Example #4	Wide	Large	Fat	Red
Example #5	Thin	Small	Slim	Blue
Example #6	Thin	Large	Fat	Blue
Example #7	Wide	Small	Fat	Blue
Example #8	Thin	Small	Slim	Blue

Unfortunately, your textbook uses an inconsistent notation to refer to values of attributes. In Chap. 13, values of attributes (= random variables) are lower-case (see Section 13.2.2). In Chap. 18, values of attributes are upper-case (see Fig. 18.3). Here, since we are in the machine learning part of the course, we will follow Chap. 18 and use upper-case values of attributes. Please do not be confused.

Recall that Bayes' rule allows you to rewrite the conditional probability of the class given the attributes as the conditional probability of the attributes given the class. As usual, α is a normalizing constant that makes the likelihoods (unnormalized probabilities) sum to one. Thus, we may ignore the repeated denominator **P**(Fins, Tail, Body), because it is constant for all classes. Using Bayes' Rule, we rewrite: **P**(Class **|** Fins, Tail, Body) = α **P**(Fins, Tail, Body **|** Class) P(Class)

3.a. (2 pts) Now assume that the attributes (Fins, Tail, and Body) are conditionally independent given the Class. Rewrite the expression above, using this assumption of conditional independence (i.e., rewrite it as a Naïve Bayes Classifier expression).

α P(Fins, Tail, Body | Class) P(Class) = α

3.b. (4 pts total; −1 for each wrong answer, but not negative) Fill in numerical values for the following expressions. Leave your answers as <u>simplified</u> common fractions (e.g., 1/4, 3/5).

P(Class=Red)= ______________ P(Class=Blue)= ______________

P(Fins=Thin | Class=Red)=______________ P(Fins=Thin | Class=Blue)=______________

P(Fins=Wide | Class=Red)=______________ P(Fins=Wide | Class=Blue)=______________

P(Tail=Large | Class=Red)=______________ P(Tail=Large | Class=Blue)=______________

P(Tail=Small | Class=Red)=______________ P(Tail=Small | Class=Blue)=______________

P(Body=Slim | Class=Red)=______________ P(Body=Slim | Class=Blue)=______________

P(Body=Fat | Class=Red)=______________ P(Body=Fat | Class=Blue)= ______________

3.c. (4 pts total, 2 pts each) Consider a new example **(Fins=Wide ^ Tail=Large ^ Body=Slim)**. Write these class probabilities as the product of α and common fractions from above. <u>You do not need to produce an actual final number; only an expression that will evaluate to the right answer.</u>

3.c.i. (2 pts) P(Class=Red | Fins=Wide ^ Tail=Large ^ Body=Slim)

= ____ α ______________________________

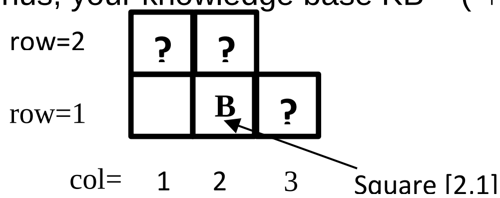
3.c.ii. (2 pts) P(Class=Blue | Fins=Wide ^ Tail=Large ^ Body=Slim)

= ____ α ______________________________

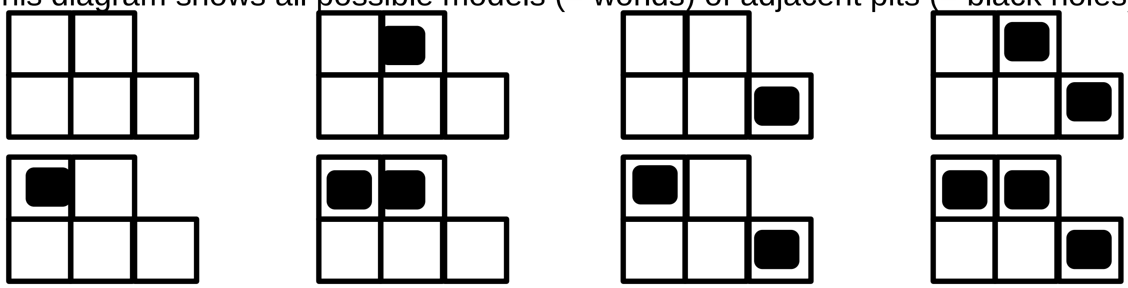
4. (10 pts total, 2 pts each) WUMPUS WORLD MODELS.
Recall that a knowledge base KB entails a sentence S (written KB |= S) just in case the set of models that make the knowledge base true is a subset of the models that make S true (a model is a possible world). If this condition holds, it is impossible for KB to be true and S to be false. In such a case, S must be true in all worlds in which KB is true.

This question will concern only breezes and pits. *<u>Squares next to pits are breezy, and breezy squares are next to squares with pits.</u>* We ignore the wumpus, gold, etc.

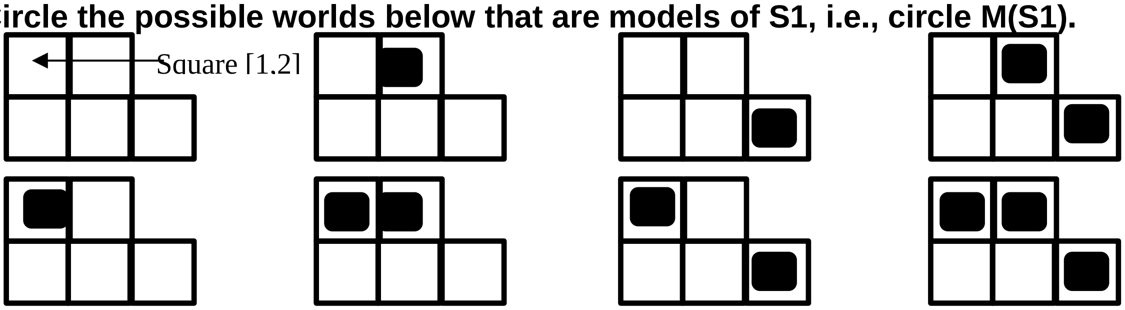
Your agent did not detect a breeze at square [1,1] **(column, row).** Square [2,1] has a breeze. Thus, your knowledge base KB = ($\neg$ B_1,1) $\wedge$ (B_2,1), where B = Breeze.



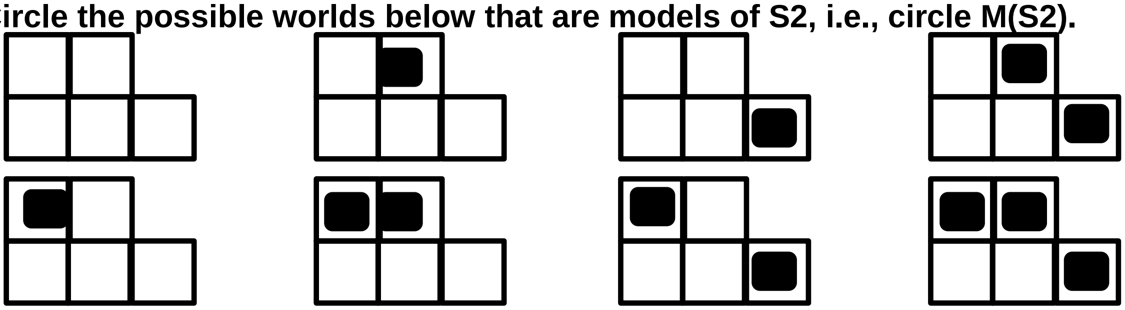
This diagram shows all possible models (= worlds) of adjacent pits (= black holes):



4.a. Circle the possible worlds above that are models of KB, i.e., circle M(KB).
4.b. Consider ONLY the sentence S1 = "Square [1,2] does not have a pit."
Circle the possible worlds below that are models of S1, i.e., circle M(S1).



4.c. Does KB |= S1? (Y = yes, N = no) ____________
4.d. Consider ONLY the sentence S2 = "Square [2,2] does not have a pit."
Circle the possible worlds below that are models of S2, i.e., circle M(S2).



4.e. Does KB |= S2? (Y = yes, N = no) ____________

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

5. (10 pts total) ONE FISH, TWO FISH, RED FISH, BLUE FISH. Decision Tree Classifiers. (With apologies to Dr. Suess.) You are a robot in the aquarium section of a pet store, and must learn to discriminate Red fish from Blue fish. Unfortunately, your vision sensors are in Black & White, but Red fish have the same gray-scale tone as Blue fish. So, you must learn to discriminate them by body parts. You choose to learn a Decision Tree classifier. You are given the following examples:

Example	Fins	Tail	Body	Class
Example #1	Thin	Small	Slim	Red
Example #2	Wide	Large	Slim	Red
Example #3	Thin	Large	Slim	Red
Example #4	Wide	Small	Medium	Red
Example #5	Thin	Small	Medium	Blue
Example #6	Wide	Large	Fat	Blue
Example #7	Thin	Large	Fat	Blue
Example #8	Wide	Small	Fat	Blue

Unfortunately, your textbook uses an inconsistent notation to refer to values of attributes. In Chap. 13, values of attributes (= random variables) are lower-case (see Section 13.2.2). In Chap. 18, values of attributes are upper-case (see Fig. 18.3). Here, since we are in the machine learning part of the course, we will follow Chap. 18 and use upper-case values of attributes. Please do not be confused.

5.a. (4 pts) Which attribute would information gain choose as the root of the tree?

___________________________.

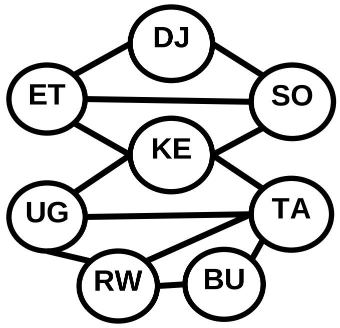
5.b. (2 pts) Draw the decision tree that would be constructed by recursively applying information gain to select roots of sub-trees, as in the Decision-Tree-Learning algorithm.

Classify these new examples as Red or Blue using your decision tree above.

5.c. (2 pts) What class is [Fins=Thin, Tail=Small, Body=Fat]? _____________________

5.d. (2 pts) What class is [Fins=Wide, Tail=Large, Body=Medium]? ________________

6. (10 points total, 2 pts each) CONSTRAINT SATISFACTION PROBLEMS.



BU = Burundi
DJ = Djibouti
ET = Ethiopia
KE = Kenya
RW = Rwanda
SO = Somalia
TA = Tanzania
UG = Uganda

You are a map-coloring robot assigned to color this East Africa map. Adjacent regions must be colored a different color (R=Red, B=Blue, G=Green). The constraint graph is shown.

6.a. (2 pts total) FORWARD CHECKING. Variable KE just now has been assigned value G, as shown. Cross out all values that would be eliminated by Forward Checking.

BU	DJ	ET	KE	RW	SO	TA	UG
R G B	R G B	R G B	G	R G B	R G B	R G B	R G B

6.b. (2 pts total) ARC CONSISTENCY.
Variables KE and UG have been assigned values, as shown, but no constraint propagation has been done. Cross out all values that would be eliminated by Arc Consistency (AC-3 in your book).

BU	DJ	ET	KE	RW	SO	TA	UG
R G B	R G B	R G B	G	R G B	R G B	R G B	R

6.c. (2 pts total) MINIMUM-REMAINING-VALUES HEURISTIC. Consider the assignment below. TA is assigned and constraint propagation has been done. List all unassigned variables that might be selected by the Minimum-Remaining-Values (MRV) Heuristic:____________________________.

BU	DJ	ET	KE	RW	SO	TA	UG
R B	R G B	R G B	R B	R B	R G B	G	R B

6.d. (2 pts total) DEGREE HEURISTIC. Consider the assignment below. (It is the same assignment as in problem 6.c above.) TA is assigned and constraint propagation has been done. Ignore MRV. List all unassigned variables that might be selected by the Degree Heuristic: ________________________

BU	DJ	T	KE	RW	SO	TA	UG
R B	R G B	R G B	R B	R B	R G B	G	R B

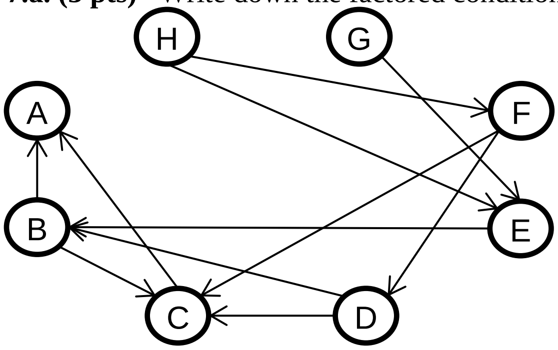
6.e. (2 pts total) MIN-CONFLICTS HEURISTIC. Consider the complete but inconsistent assignment below. UG has just been selected to be assigned a new value during local search for a complete and consistent assignment. What new value would be chosen below for UG by the Min-Conflicts Heuristic?______________________________.

BU	DJ	ET	KE	RW	SO	TA	UG
B	G	G	G	G	B	B	**?**

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

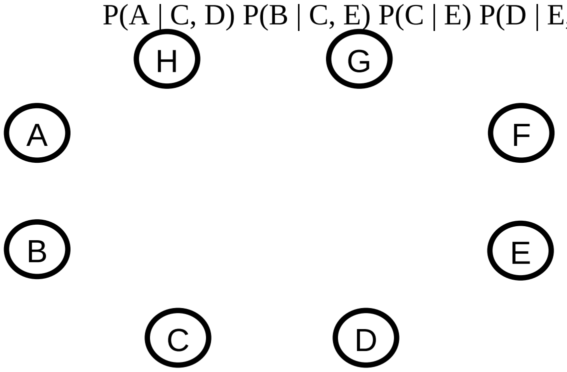
7. (10 pts total) BAYESIAN NETWORKS.

7.a. (3 pts) Write down the factored conditional probability expression corresponding to this Bayesian Network:

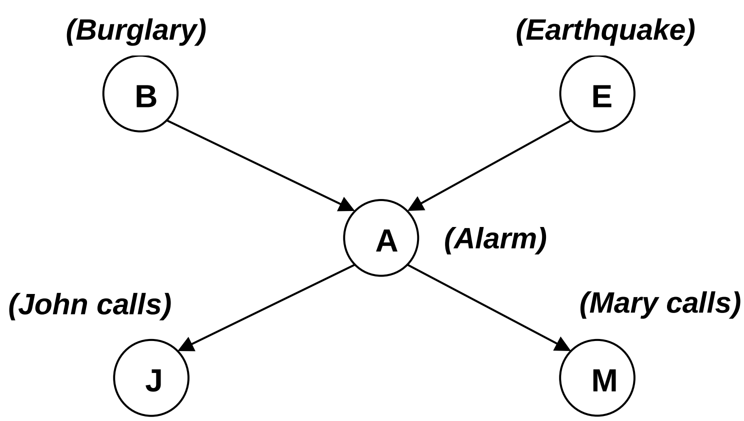


7.b. (3 pts) Draw the Bayesian Network corresponding to this factored conditional probability expression:

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

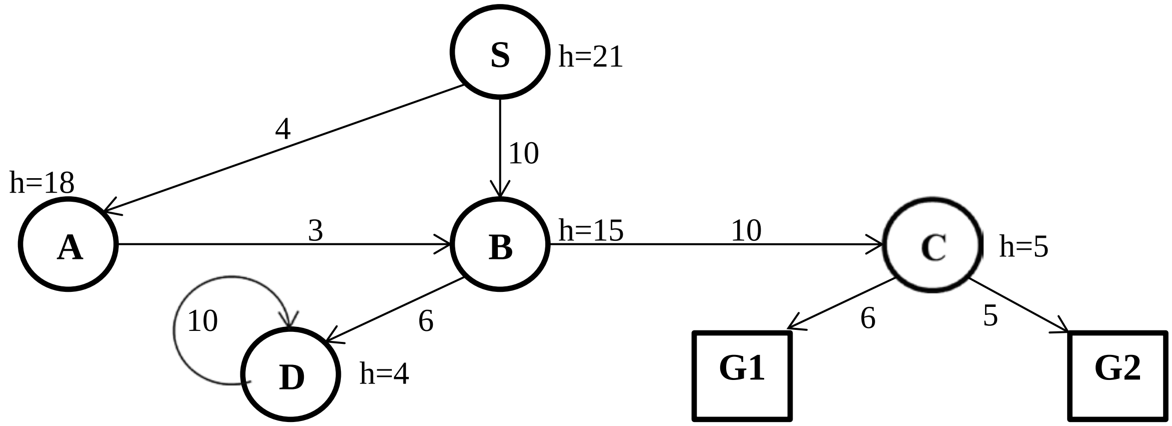
A	P(M)
t	.70
f	.01

Write down an expression that will evaluate to P(J=f $\wedge$ M=t $\wedge$ A=t $\wedge$ B=t $\wedge$ E=f). **Express your answer as a series of numbers (numerical probabilities) separated by multiplication symbols.** You do not need to carry out the multiplication to produce a single number (probability). **SHOW YOUR WORK.**

P(J=f $\wedge$ M=t $\wedge$ A=t $\wedge$ B=t $\wedge$ E=f)
=

8. (10 pts total, 2 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). The successors of each node are indicated by the arrows out of that node. **(Note: D is a successor of itself).** As usual, successor nodes are returned in left-to-right order. (The successor nodes of S are A,B; and the successor nodes of B are D,C; in those node orders).

The start node is S and there are two goal nodes, G1 and G2. For each search strategy below, indicate **(1) the order** in which nodes are expanded, and **(2) the path and cost** to the goal that was found, if any. Write "None" for the path and cost if the goal was not found. The first one is done for you, as an example.



8.a. (example) DEPTH-FIRST SEARCH:

8.a.i Order of expansion: S A B D D D D ... ____________________

8.a.ii Path to goal found: None ____________________ Cost of path found: None

8.b. (2 pts) BREADTH-FIRST SEARCH:

8.b.i Order of expansion: ____________________

8.b.ii Path to goal found: ____________________ Cost of path found: ________

8.c. (2 pts) ITERATIVE DEEPENING SEARCH:

8.c.i Order of expansion: ____________________

8.c.ii Path to goal found: ____________________ Cost of path found: ________

8.d. (2 pts) UNIFORM COST SEARCH:

8.d.i Order of expansion: ____________________

8.d.ii Path to goal found: ____________________ Cost of path found: ________

8.e. (2 pts) GREEDY BEST FIRST SEARCH:

8.e.i Order of expansion: ____________________

8.e.ii Path to goal found: ____________________ Cost of path found: ________

8.f. (2 pts) A* SEARCH:

8.f.i Order of expansion: ____________________

8.f.ii Path to goal found: ____________________ Cost of path found: ________

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

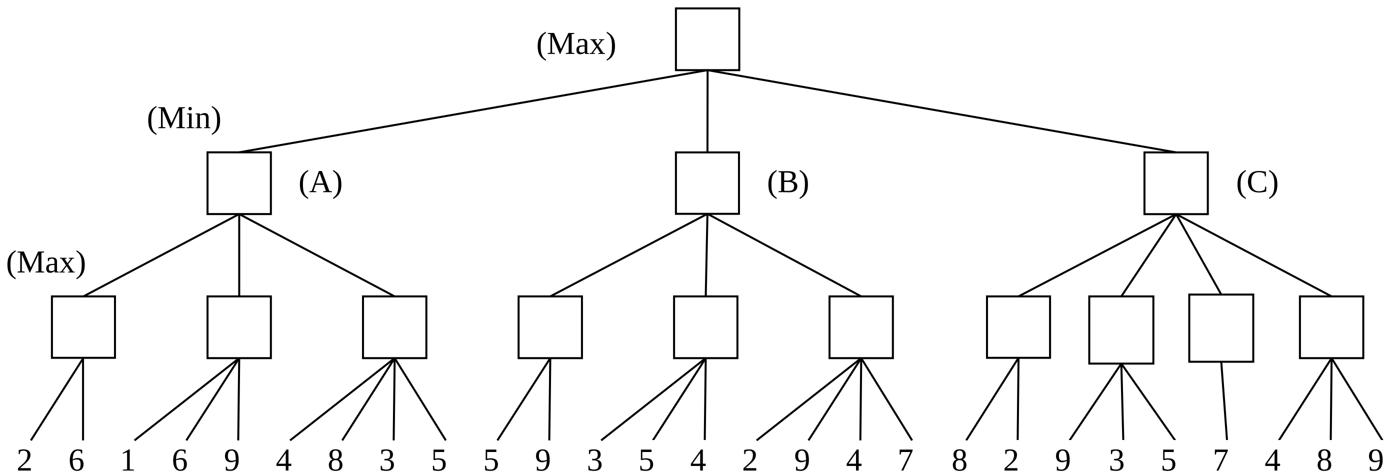
9. (10 pts total, -1 pt for each error, but not negative) GAME (ADVERSARIAL) SEARCH.

9.a. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.
The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is **Max**'s turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

9.a.i. Fill in each blank square with the proper mini-max search value.

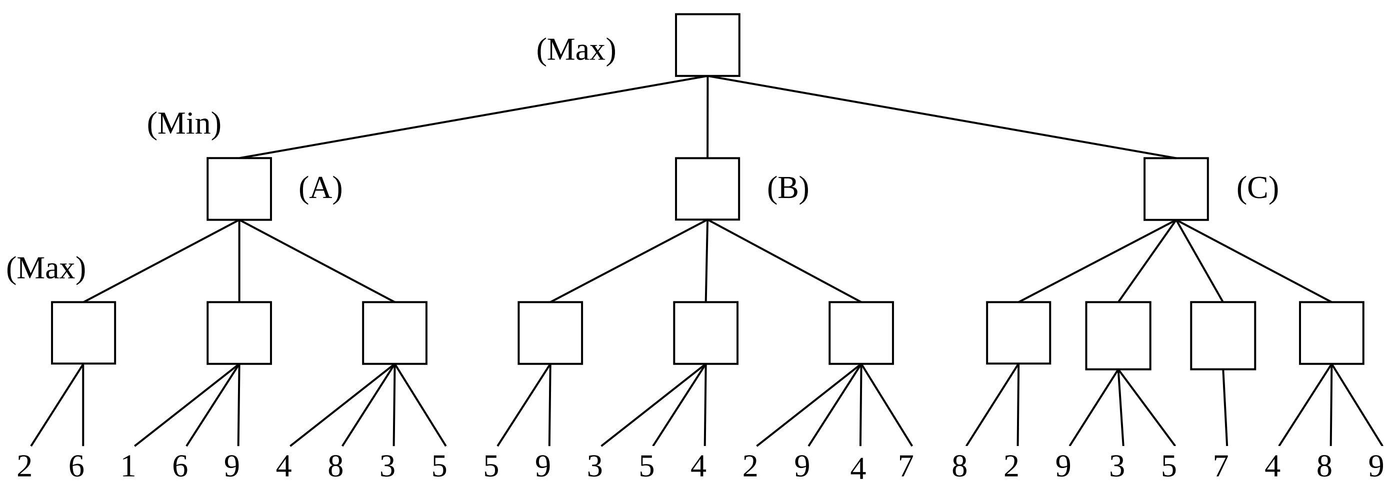
9.a.ii. What is the best move for Max? (write A, B, or C) _____________

9.a.iii. What score does Max expect to achieve? __________________

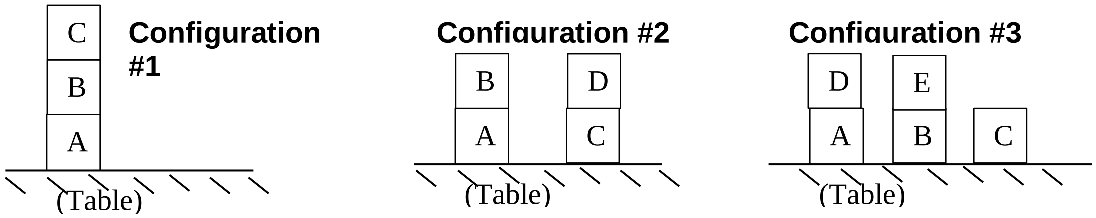


9.b. (5 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (9.a). You do not need to indicate the branch node values again.

<u>Cross out each leaf node that will be pruned by Alpha-Beta Pruning.</u>



10. (10 pts total, 1 pt each) FOPC KNOWLEDGE ENGINEERING IN THE TOY BLOCKS WORLD.
You are a Knowledge Engineer assigned to the Toy Blocks World, which involves directing a controller for a robot arm that stacks children's toy blocks one atop another, or on a table, into a desired configuration. For example, you are concerned with configurations such as these:



Here, we wish to describe only static (unmoving) configurations that eventually will become goals (targets) in the Toy Blocks World. A separate module, which is not your concern, later will move the robot arm to achieve these goals. Here, you need only to describe correctly in FOPC the static goal (target) configurations.

Use the primitive predicate "Stacked(x, y)" to mean that "Block x is stacked directly on top of block y." Below, we also define new predicates "Clear(x)", "OnTable(x)", and "HigherThan(x,y)", which may be used elsewhere. <u>Assume that all objects in the world are blocks</u>, i.e., there is no need for Block(x) guard predicates.

For each English statement below, write the best match letter chosen from the FOPC sentences that follow at the bottom of the page. The first one is done for you as an example.

10.a. (example) _______A______ Assert that "Block x is stacked on block y" implies "Block y is not stacked on block x."

10.b. (1 pt) __________ Define a predicate "Clear(x)" to mean that no block y is stacked on block x.

10.c. (1 pt) __________ Define a predicate "OnTable(x)" to mean that block x is on the table, i.e., not on any block y.

10.d. (1 pt) __________ Define a predicate "Above(x, y)" to mean that x is above y in a stack that includes both x and y.

10.e. (1 pt) __________ State that at least one block must be clear, i.e., at least one block must have no other block stacked upon it. You may use the Clear(x) predicate defined in (10.b) above

10.f. (1 pt) __________ State that at least one block must be on the table, i.e., at least one block must not be stacked on any other block. You may use the OnTable(x) predicate defined in (10.c) above.

10.g. (1 pt) __________ Define a predicate "HigherThan(x, y)" to mean that x is at a higher altitude above the table than is y, even though x and y may be in different stacks. You may use the OnTable(x) predicate defined in (10.c) above.

10.h. (1 pt) __________ Assert that "Block x is above block y" implies "Block x is higher than block y."

10.i. (1 pt) __________ Describe Configuration #1 in FOPC.

10.j. (1 pt) __________ Describe Configuration #2 in FOPC.

10.k. (1 pt) __________ Describe Configuration #3 in FOPC.

A. $\forall x, y\ \text{Stacked}(x, y) \Rightarrow \neg\text{Stacked}(y, x)$
B. $\exists x\ \text{Clear}(x)$
C. $\forall x, y\ \text{Above}(x, y) \Rightarrow \text{HigherThan}(x, y)$
D. $\text{OnTable}(A) \wedge \text{Stacked}(B,A) \wedge \text{Stacked}(C,B) \wedge \text{Clear}(C)$
E. $\forall x, y\ \text{OnTable}(x) \Leftrightarrow \neg\text{Stacked}(x, y)$
F. $\exists x\ \text{OnTable}(x)$
G. $\text{OnTable}(A) \wedge \text{Stacked}(B,A) \wedge \text{Clear}(B) \wedge \text{OnTable}(C) \wedge \text{Stacked}(D,C) \wedge \text{Clear}(D)$
H. $\forall x, y\ \text{Above}(x, y) \Leftrightarrow [\ \text{Stacked}(x, y) \vee (\ \exists z\ \text{Stacked}(z, y) \wedge \text{Above}(x, z)\ )\ ]$
I. $\forall x, y\ \text{Clear}(x) \Leftrightarrow \neg\text{Stacked}(y, x)$
J. $\text{OnTable}(A) \wedge \text{Stacked}(D,A) \wedge \text{Clear}(D) \wedge \text{OnTable}(B) \wedge \text{Stacked}(E,B) \wedge \text{Clear}(E) \wedge \text{OnTable}(C) \wedge \text{Clear}(C)$
K. $\forall x, y\ \text{HigherThan}(x, y)$
$\Leftrightarrow [\ (\neg\text{OnTable}(x) \wedge \text{OnTable}(y))\ \vee (\ \exists w,z\ \text{Stacked}(x,w) \wedge \text{Stacked}(y,z) \wedge \text{HigherThan}(w,z)\ )\ ]$

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