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.

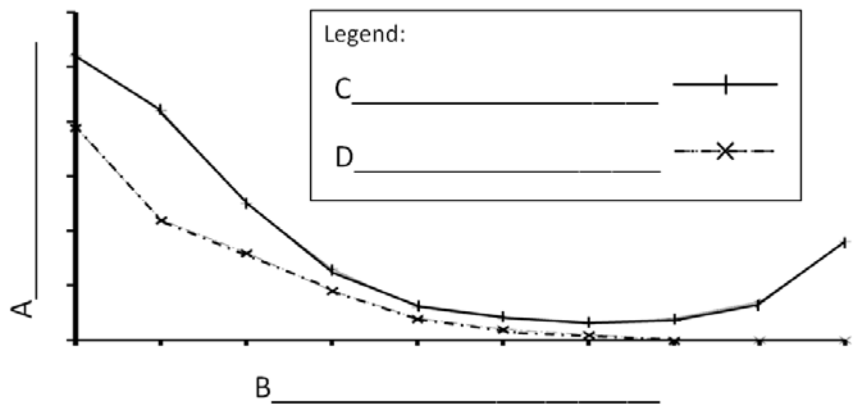
9. (10 pts total, -1 pt for each error, but not negative) GAME (ADVERSARIAL) SEARCH.

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.

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

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

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

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

See Fig. 18.9

Typically, training set error (D) decreases with increasing tree size (B), whereas minimum validation set error (C) requires model complexity to be in a "sweet spot". Both C and D are error rates (A).

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) ____**C**____ (Write A, B, C, or D.) Which c[...] perform[...] when applied to unseen data?

Best generalization performance = smallest error on the testing data (validation set).

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

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

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

Underfitting: Classifier is not complex enough to achieve low error rates on both training and testing set. A has the largest overall error, thus A is underfitting the most.

Overfitting: Classifier performs far better on training set than on testing set (limited ability to generalize to unseen data). B has the largest gap between training and testing error, thus B is overfitting the most.

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

Linear perceptron fails because data is not linearly separable.

Both DT and SVM support non-linear decision boundaries.

For SVMs, this is just textbook Fig. 18.31.

DTs are DNF-complete, so can separate any consistent training set (but may need a tree of exponential size!!).

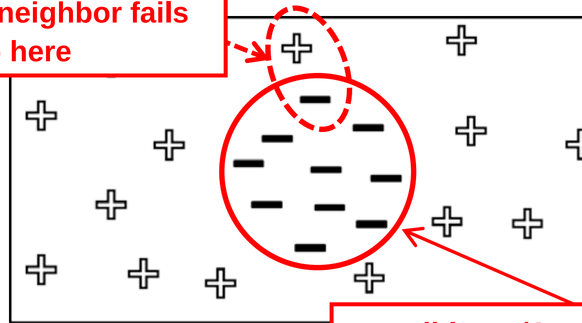
1.c.i.
(1 pt)

☐ Linear Perceptron

☐ Nearest Neighbor

☒ Decision Tree

☒ Support Vector Machine



Nearest neighbor fails (at least) here

Possible DT/SVM decision boundary (DT: picture a pixelated circle).

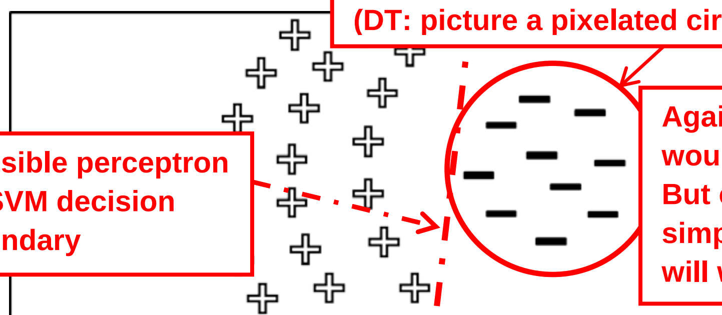
1.c.ii.
(1 pt)

☒ Linear Perceptron

☒ Nearest Neighbor

☒ Decision Tree

☒ Support Vector Machine



Possible perceptron or SVM decision boundary

Again a circle would do the trick. But careful! A simple straight line will work, too.

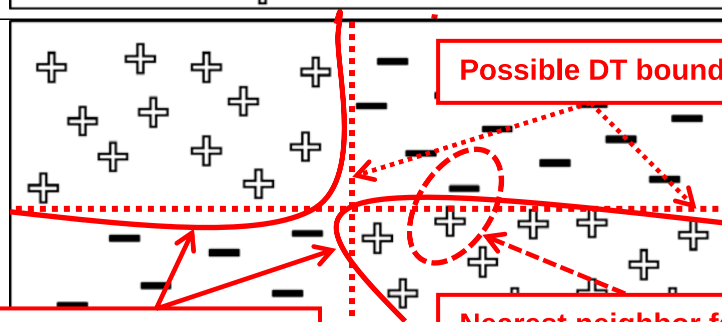
1.c.iii.
(1 pt)

☐ Linear Perceptron

☐ Nearest Neighbor

☒ Decision Tree

☒ Support Vector Machine



Possible DT boundaries

Possible SVM boundaries

Nearest neighbor fails (at least) here

For question 1.c.iii. you received full credit if you answered DT or SVM or both.

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:

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

(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))

See R&N Section 7.5.2.

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)".

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

(A B C)	((¬ A) (¬ B) (¬ C))	Among them they had exactly two red fish and two blue fish.
(A B D)	((¬ A) (¬ B) (¬ D))	
(A C D)	((¬ A) (¬ C) (¬ D))	
(B C D)	((¬ B) (¬ C) (¬ D))	
((¬ A) D)	(A (¬ D))	(<=> A D)
(A B C)	((¬ A) (¬ B))	Amy, Betty, and Cindy had exactly one red fish among them.
((¬ A) (¬ C))	((¬ B) (¬ C))	
(B)		Negated goal.

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 five 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, Amy and Diane must have had red fish, while Betty and Cindy had blue fish.*

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

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

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

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

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

Resolve ________ with ________ to produce: ________

Resolve ________ with ________ to produce: ________

Resolve ________ with ________ to produce: ________

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

Other proofs are OK provided that they are correct. For example, another correct proof is:

Resolve ______B____________with ___((¬ A) (¬ B))_______________ to produce: ___(¬ A)_________

Resolve ______B____________ with ___((¬ B) (¬ C))_______________ to produce: ___(¬ C)_________

Resolve ______(¬ A)__________ with ___(A (¬ D))_________________ to produce: ___(¬ D)_________

Resolve ______(¬ A)__________ with ___(A C D)____________________ to produce: ___(C D)_________

Resolve ______(¬ C)__________ with ___(C D)______________________ to produce: ___(D)__________

Resolve ______(¬ D)__________ with ___(D)________________________ to produce: ___( )___________

Resolve __________________ with __________________________ to produce: _______________

See Sections 13.5.2 and 20.2.2.

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 ... ve the same gray-s ... u choose to learn a

This is the same as problem #2, Quiz #4, WQ 2015; and also #3 on Final Exam, WQ 2012; except Red fish replaces Dog, Blue fish replaces Cat, and attribute & value names were changed into a fish theme.

The class website, in the "Study Guides" section, states, "In particular, questions that many students missed are likely to appear again. If you missed a question, please study it carefully and learn from your mistake --- so that if it appears again, you will understand it perfectly."

Many students missed points on this question during Quiz #4 of this quarter. Consequently, it appears again on the Final Exam.

I hope that you have studied carefully your mistakes, now understand it perfectly, and now scored 100% on this problem.

Example	Fins
Example #1	Thin
Example #2	Wide
Example #3	Wide
Example #4	Wide
Example #5	Thin
Example #6	Thin
Example #7	Wide
Example #8	Thin

values of attributes. Please do not be confused.

... es' 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 ... that makes the likelihoods (unnormalized probabilities) sum to one. Thus, we ... repeated denominator **P**(Fins, Tail, Body), because it is constant for all class ... we rewrite: **P**(Class | Fins, Tail, Body) = α **P**(Fins, Tail, Body | Class) P(Clas ...

The probabilities in problem 2b are obtained by counting examples in the training set. E.g., P(Class=Red)=4/8=1/2 because 4 of the 8 examples have Class=Red. E.g., P(Fins=Thin|Class=Red)=1/4 because 1 of the 4 examples with Class=Red also has Fins=Thin.

3.a. (2 pts) Now assume that the attributes (Fins, Tail, and Body) are conditi ... the Class. Rewrite the expression above, using this assumption of condition ... rewrite it as a Naïve Bayes Classifier expression).

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

P(Fins | Class) P(Tail | Class) P(Body | Class) P(Class)

3.b. (4 pts total; −1 for each wrong answer, but not negative) Fill in nume ... following expressions. Leave your answers as simplified common fractions (e.g., 1/4, 3/8).

P(Class=Red)= 1/2 | P(Class=Blue)= 1/2

P(Fins=Thin | Class=Red)= 1/4 | P(Fins=Thin | Class=Blue)= 3/4

P(Fins=Wide | Class=Red)= 3/4 | P(Fins=Wide | Class=Blue)= 1/4

P(Tail=Large | Class=Red)= 3/4 | P(Tail=Large | Class=Blue)= 1/4

P(Tail=Small | Class=Red)= 1/4 | P(Tail=Small | Class=Blue)= 3/4

P(Body=Slim | Class=Red)= 1/2 | P(Body=Slim | Class=Blue)= 1/2

P(Body=Fat | Class=Red)= 1/2 | P(Body=Fat | Class=Blue)= 1/2

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. You do not need to produce an actual final number; only an expression that will evaluate to the right answer.

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

= α(3/4)(3/4)(1/2)(1/2) **(=9/10)**

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

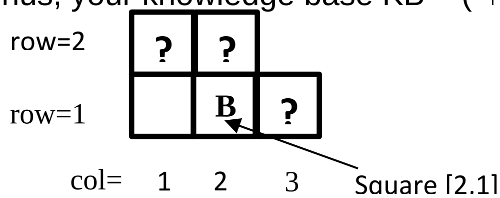
= α(1/4)(1/4)(1/2)(1/2) **(=1/10)**

You are not obliged to provide the (red) "(=9/10)" and "(=1/10)" evaluations; only the fractional products in black that precede them in the answer. The final normalized probabilities (in red) are only for your information in seeing it work.

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.

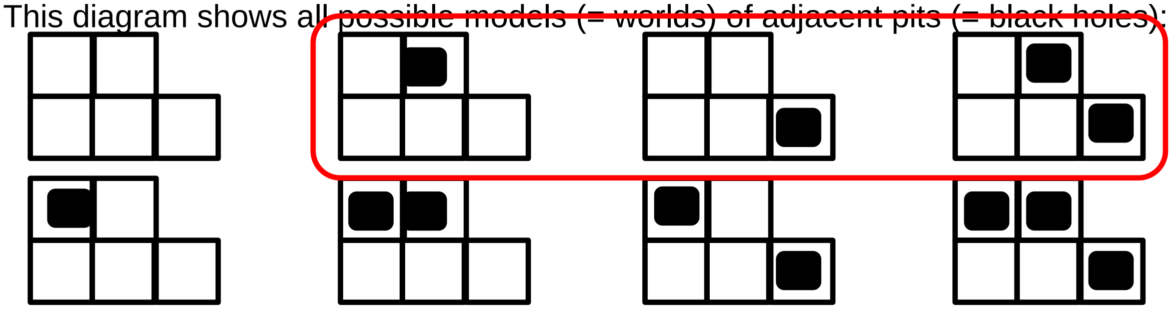
This question will concern only breezes and pits. *Squares next to pits are breezy, and breezy squares are next to squares with pits.* We ignore the wumpus, gold, etc.

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.

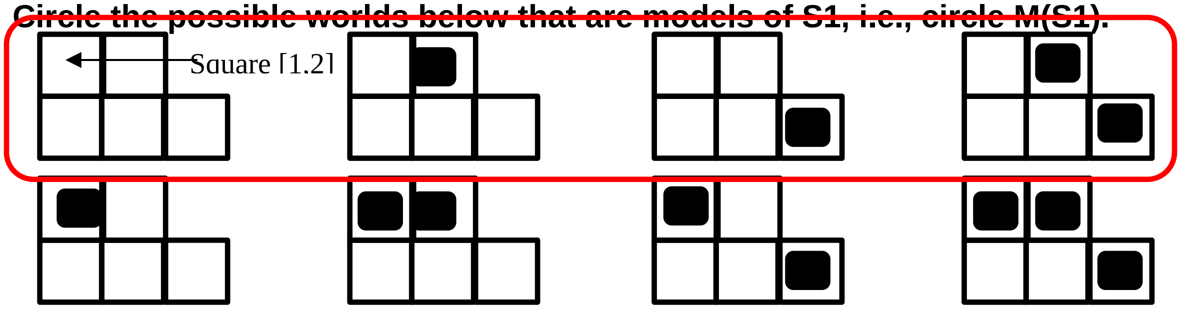


This problem is the same as #1 on WQ'2013 Mid-term. See Section 7.3 and Fig. 7.5, p. 241, of R&N; and lecture slides for Propositional Logic A and Review for Mid-term Exam.

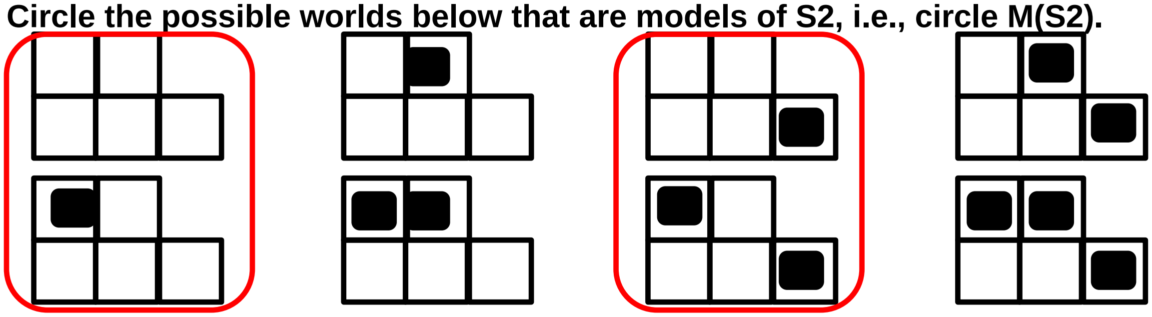
This diagram shows all possible models (= worlds) of adjacent pits (= black holes):



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) _____Y_____
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) _____N_____

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

See Section 18.3.

5. (10 pts total) ONE FISH, TWO FISH, RED FI
(With apologies to Dr. Suess.) You are a robot
learn to discriminate Red fish from Blue fish. Un
ve the same gray-scale tone as B
choose to learn a Decision Tree

This is the same as problem #1, Quiz #4, WQ 2015; and also #8 on Final Exam, FQ 2013; except Red fish replaces Oak wood, Blue fish replaces Pine wood, and attribute & value names were changed into a fish theme.

The class website, in the "Study Guides" section, states, "In particular, questions that many students missed are likely to appear again. If you missed a question, please study it caref and learn from your mistake --- so that if it appears again, yo will understand it perfectly."

Many students missed points on this question during Quiz # this quarter. Consequently, it appears again on the Final Exa

I hope that you have studied carefully your mistakes, now understand it perfectly, and now scored 100% on this proble

Example	Fins	Tail	Body	
Example #1	Thin	Small	Slim	
Example #2	Wide	Large	Slim	
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

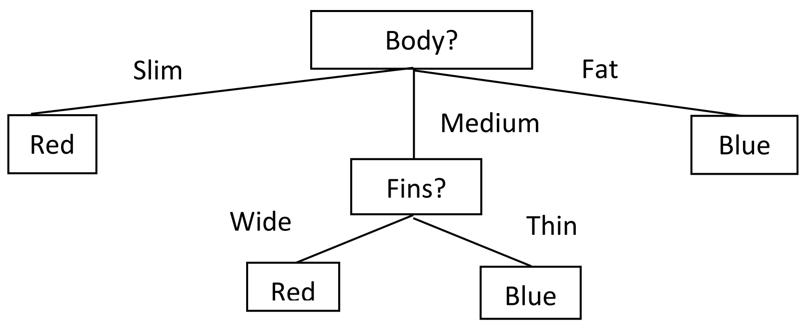
13, values of attributes (= random variables) are
lower case (see Section 13.2.2). In Chap. 18, values
o
si
c se
v

If root is Fins:
Thin = RRBB, Wide = RRBB
If root is Tail:
Small = RRBB, Large = RRBB
If root is Body:
Slim=RRR, Medium=RB, Fat=BBB
(R = Red, B = Blue)
Note that you do not need math or a calculator to answer this correctly.
Obviously, Fins and Tail do not reduce entropy, while Body does.

5.a. (4 pts) Which attribute would information gain choose

Body .

5.b. (2 pts) Draw the decision tree that would be constructe in
to select roots of sub-trees, as in the Decision-Tree-Learnin



After we choose Body for the root:
* Tail does not discriminate the two remaining unclassified examples (#4/Red & #5/Blue) because Tail for both is Small.
* However, Fins does separate them perfectly, because Fins is Wide for #4/Red and Thin for #5/Blue.

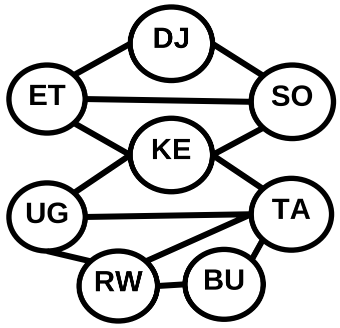
Classify these new examples as Red or Blue using your decision tree above.

Full credit if your answers are right for the tree you drew, even if the tree itself is wrong.

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

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

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

See Chapter 6.

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. **See Section 6.3.2.**

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

See Section 6.2.2.

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. Cor **See Section 6.3.1.** ent 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, KE, RW, UG** .

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 **See Section 6.3.1.** . (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: **ET, KE, SO**

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

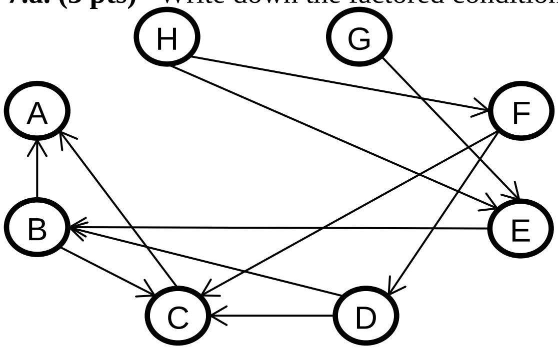
6.e. (2 pts total) MIN-CONFLICTS HEURISTIC. Consider the com **See Section 6.4.** tent assignment below. UG has just been selected to be assigned a new value during local search for a complete and consistent assignment. What new value would be chosen below for UG by the Min-Conflicts Heuristic? **R** .

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.

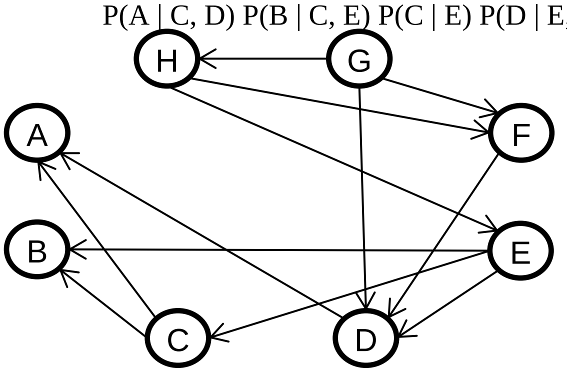
7.a. (3 pts) Write down the factored conditional probability expression corresponding to this Bayesian Network:



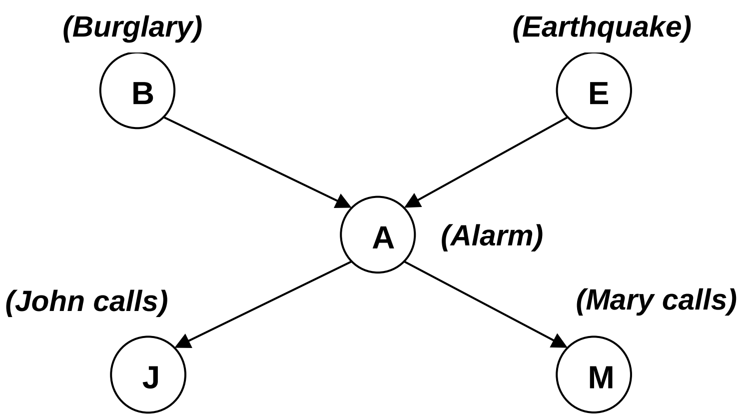
P(A | B, C) P(B | D, E) P(C | B, D, F) P(D | F) P(E | G, H) P(F | H) P(G) P(H)

7.b. (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 ∧ M=t ∧ A=t ∧ B=t ∧ E=f). **<u>Express your answer as a series of numbers (numerical probabilities) separated by multiplication symbols.</u>** You do not need to carry out the multiplication to produce a single number (probability). **<u>SHOW YOUR WORK.</u>**

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

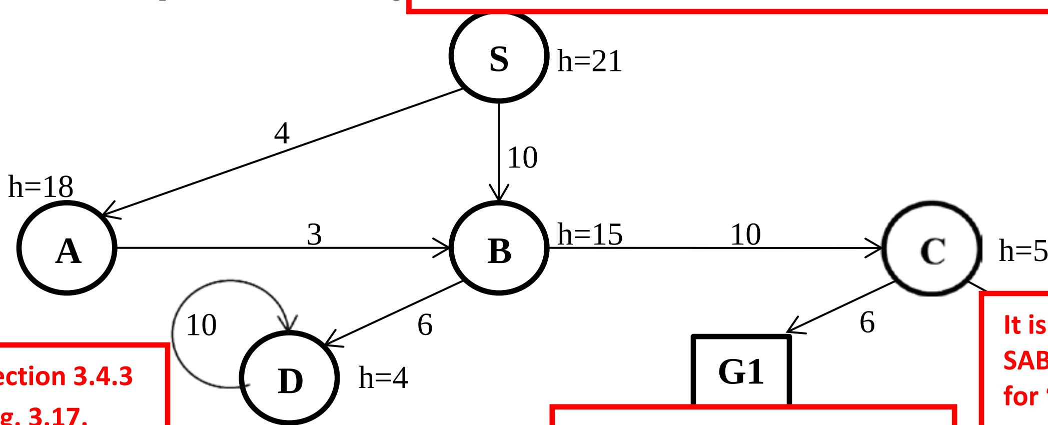
8. (10 pts total, 2 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (do not remember visited nodes, so repeated nodes are possible). It is not a tree, but pretend you don't know that. Step costs are given next to each arc, and heuristic values are given next to each node (as h=x). 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

The start node is S and there are tw … e

(1) the order in which nodes are expanded

Write "None" for the path and cost if the g

Please see the lecture slides for Uninformed Search, topic "When to do Goal-Test? When generated? When popped?" for clarification about exactly what to do in practical cases.



See Section 3.4.3 and Fig. 3.17.

It is OK if you wrote SABDDD... instead of None for "Path found." It is OK if you said N/A for "Cost of path found," or left it blank.

H-FIRST SEARCH:

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

DFS can get caught in loops during Tree Search (= do not remember visited nodes).

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

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

See Section 3.4.1 and Fig. 3.11.

on: S A B B D C G1

BFS does the Goal-test before the child is pushed onto the queue. The goal G1 is found when C is expanded.

8.b.ii Path to goal found: S B C G1 Cost of path found: 26

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

See Sections 3.4.4-5 and Figs. 3.18-19.

S S A B S A B B D C G1

IDS does the Goal-test before the child is pushed onto the queue. The goal G1 is found when C is expanded.

S B C G1 Cost of path found: 26

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

UCS does Goal-test when node is popped off

See Section 3.4.2 and Fig. 3.14.

on: S A B B D D C C G2

d: S A B C G2 Cost of path found: 22

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

GBFS can get caught in loops during Tree Search (= do not remember visited nodes). The heuristic value at node D (h=4) is lower than any other heuristic value on the queue.

See Section 3.5.1 and Fig. 3.23.

on: S B D D D D ...

d: None Cost of path found: None

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

A* does Goal-test when node is popped off

See Section 3.5.2 and Figs. 3.24-25.

n: S A B D C G2

d: S A B C G2 Cost of path found: 22

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

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

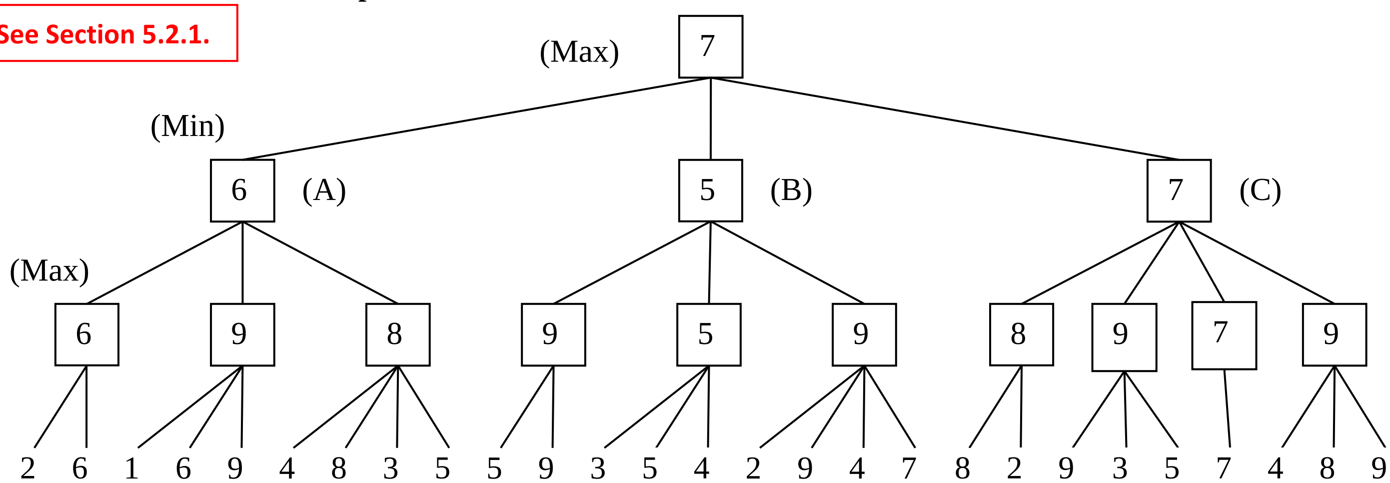
9.a. (5 pts total, -1 pt for each error, but not negative) MINI-MAX SEARCH IN GAME TREES.
The game tree below illustrates a position reached in the game. Process the tree left-to-right. It is **Max**'s turn to move. At each leaf node is the estimated score returned by the heuristic static evaluator.

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) C

9.a.iii. What score does Max expect to achieve? 7

See Section 5.2.1.

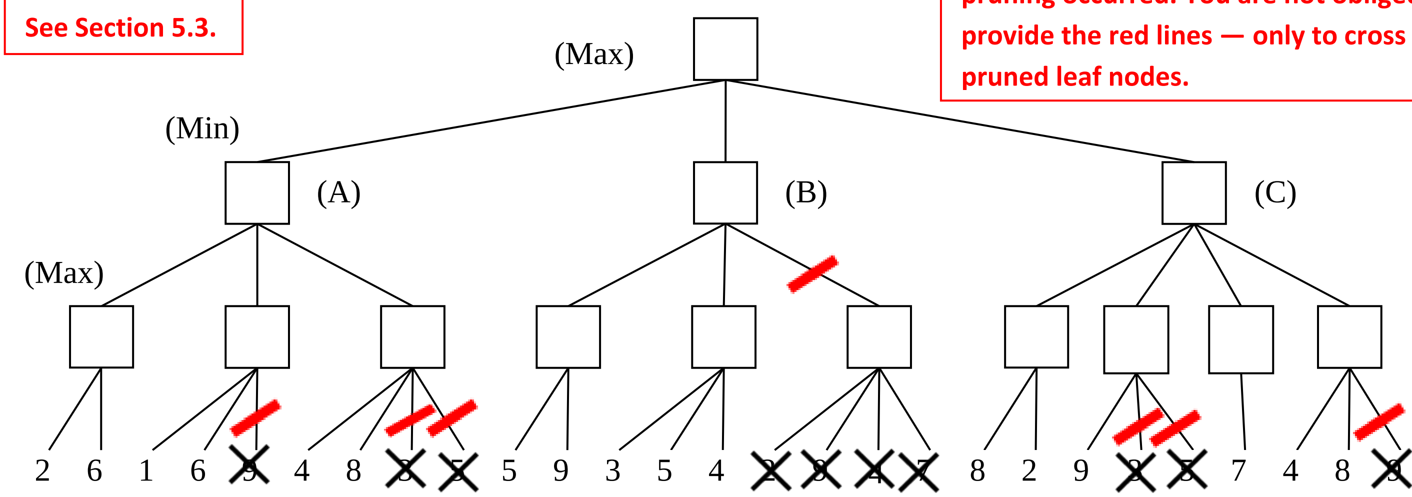


9.b. (5 pts total, -1 pt for each error, but not negative) ALPHA-BETA PRUNING. Process the tree left-to-right. This is the same tree as above (9.a). You do not need to indicate the branch node values again.

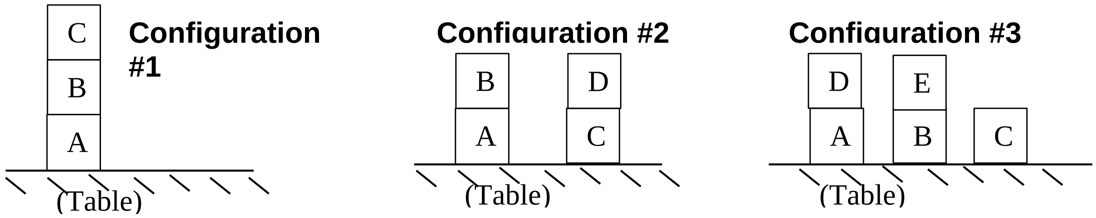
<u>Cross out each leaf node that will be pruned by Alpha-Beta Pruning.</u>

Red lines indicate where in the tree pruning occurred. You are not obliged to provide the red lines — only to cross out pruned leaf nodes.

See Section 5.3.



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. Assume that all objects in the world are blocks, 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) ____I____ Define a predicate "Clear(x)" to mean that no block y is stacked on block x.

10.c. (1 pt) ____E____ 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) ____H____ 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) ____B____ 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) ____F____ 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) ____K____ 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) ____C____ Assert that "Block x is above block y" implies "Block x is higher than block y."

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

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

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

A. $\forall x, y\ Stacked(x, y) \Rightarrow \neg Stacked(y, x)$

B. $\exists x\ Clear(x)$

C. $\forall x, y\ Above(x, y) \Rightarrow HigherThan(x, y)$

D. $OnTable(A) \wedge Stacked(B,A) \wedge Stacked(C,B) \wedge Clear(C)$

E. $\forall x, y\ OnTable(x) \Leftrightarrow \neg Stacked(x, y)$

F. $\exists x\ OnTable(x)$

G. $OnTable(A) \wedge Stacked(B,A) \wedge Clear(B) \wedge OnTable(C) \wedge Stacked(D,C) \wedge Clear(D)$

H. $\forall x, y\ Above(x, y) \Leftrightarrow [\ Stacked(x, y) \vee (\ \exists z\ Stacked(z, y) \wedge Above(x, z)\ )\ ]$

I. $\forall x, y\ Clear(x) \Leftrightarrow \neg Stacked(y, x)$

J. $OnTable(A) \wedge Stacked(D,A) \wedge Clear(D) \wedge OnTable(B) \wedge Stacked(E,B) \wedge Clear(E) \wedge OnTable(C) \wedge Clear(C)$

K. $\forall x, y\ HigherThan(x, y)$
$\Leftrightarrow [\ (\neg OnTable(x) \wedge OnTable(y))\ \vee (\ \exists w,z\ Stacked(x,w) \wedge Stacked(y,z) \wedge HigherThan(w,z)\ )\ ]$

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