CS-171, Intro to A.I. — Mid-term Exam — Fall Quarter, 2012

YOUR NAME AND ID NUMBER: ____

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 seven pages, as numbered 1-7 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, 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 turn it in with your exam.

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

1. (10 pts total, -3 for each error, but not negative) TASK ENVIRONMENT.

2. (10 pts total, -2 for each error, but not negative) RESOLUTION THEOREM PROVING.

3. (15 pts total, 3 pts each) CONSTRAINT SATISFACTION PROBLEMS.

4. (15 pts total, 3 pts each) STATE-SPACE SEARCH.

5. (15 pts max, -2 for each error, but not negative) MINIMAX WITH ALPHA-BETA PRUNING.

6. (10 pts total, -1 each error, but not negative) CONJUNCTIVE NORMAL FORM (CNF).

7. (10 pts total, 2 pts each) RESOLUTION.

8. (15 pts total, -1 each wrong answer, but not negative) SEARCH PROPERTIES.

1. (10 pts total, -3 for each error, but not negative) TASK ENVIRONMENT. Your book defines a task environment as a set of four things, with the acronym PEAS. Fill in the blanks with the names of the PEAS components.

Performance (measure) Environment Actuators Sensors

2. (10 pts total, -2 for each error, but not negative) RESOLUTION THEOREM PROVING. You are engaged in Knowledge Engineering for the Wumpus Cave. You have interviewed an expert on the Wumpus Cave who told you, among other things, "A breeze in square (1,1) is equivalent to a pit in square (1,2) or a pit in square (2,1)." You translated this into propositional logic as, "(B11 ⇔ P12 ∨ P21)," and then into Conjunctive Normal Form as "(¬B11 ∨ P12 ∨ P21) ∧ (¬P12 ∨ B11) ∧ (¬P21 ∨ B11)."

Now it is time for the first "live" test of your system. An agent has been lowered down into the Wumpus cave, and reports back by radio, "Square (1,1) does not have a breeze." You translate this knowledge into propositional logic as "(¬B11)" and add it to your knowledge base.

Next your system is asked to perform inference. The agent asks by radio, "Is it true that square (1,2) does not have a pit AND that square (2,1) does not have a pit?" You translate this query into propositional logic as the goal sentence "(¬P12) ∧ (¬P21)." You form the negated goal as "(P12∨ P21)." Now your knowledge base plus negated goal is:

(¬B11 ∨ P12 ∨ P21)
(¬P12 ∨ B11)
(¬P21 ∨ B11)
(¬B11)
(P12∨ P21)

Run resolution on this knowledge base until you produce the null clause, "( )", thereby proving that the goal sentence is true. The shortest proof I know of is only four lines, including the first example line. It is OK to use more lines, if your proof is correct. SHOW YOUR WORK.

Repeatedly choose two clauses, write one clause in the first blank space on a line, and the other clause in the second. Apply resolution to them. Write the resulting clause in the third blank space, and insert it into the knowledge base. The first one is done for you as an example.

Resolve (¬P12 ∨ B11) and (¬B11) to give (¬P12).

Resolve (¬P21 ∨ B11) and (¬B11) to give (¬P21).

Resolve (P12∨ P21) and (¬P12) to give (P21).

Resolve (P21) and (¬P21) to give ( ).

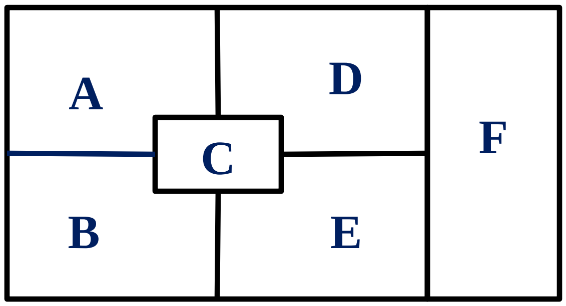
Resolve ____ .

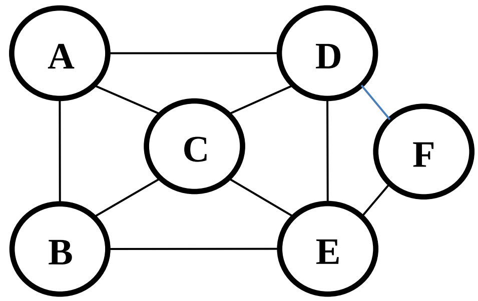
Resolve ____ .

Other proofs are OK as long as they are correct. For example, another proof is:
Resolve (¬P12 ∨ B11) and (¬B11) to give (¬P12).
Resolve (¬P12) and (P12∨ P21) to give (P21).
Resolve (P21) and (¬P21 ∨ B11) to give (B11).
Resolve (B11) and (¬B11) to give ( ).

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

3. (15 pts total, 3 pts each) CONSTRAINT SATISFACTION PROBLEMS. This problem asks about the Map Coloring Problem. Each region must be colored one of Red (R), Green (G), or Blue (B). Neighboring regions must be a different color. The map (left) and constraint graph (right) are below.





3a. (3 pts) FORWARD CHECKING. Consider the partial assignment below. Variable E has been assigned value R as shown. Cross out all values that would be eliminated by Forward Checking (FC) after the assignment to variable E.

A	B	C	D	E	F
R G B	~~R~~ G B	~~R~~ G B	~~R~~ G B	R	~~R~~ G B

See Section 6.3.2.

3b. (3 pts) ARC CONSISTENCY. Consider the partial assignment below. Variables A and B have been assigned values as shown. Cross out all other values that would be eliminated by Arc Consistency (AC, also called AC-3 in your book).

A	B	C	D	E	F
R	G	~~R~~ ~~G~~ B	~~R~~ G ~~B~~	R ~~G~~ ~~B~~	~~R~~ ~~G~~ B

See Section 6.2.2.

3c. (3 pts) MINIMUM-REMAINING-VALUES HEURISTIC. Consider the partial assignment below. Variable A is already assigned value R, and Arc Consistency is already done. List all unassigned variables that might possibly be selected by the Minimum-Remaining-Values (MRV) Heuristic: **B, C, D** .

A	B	C	D	E	F
R	G B	G B	G B	R G B	R G B

3d. (3 pts) DEGREE HEURISTIC. Consider the partial assignment below. (It is the same assignment as in problem 3c above.) List all unassigned variables that might possibly be selected by the Degree Heuristic: **E** .

A	B	C	D	E	F
R	G B	G B	G B	R G B	R G B

See Section 6.3.1.

3e. (3 pts) MIN-CONFLICTS HEURISTIC. Consider the complete but inconsistent assignment below. E has just now been selected to be assigned a new value. List all new values that might be chosen below for E by the Min-Conflicts Heuristic? **R, G** .

A	B	C	D	E	F
R	G	B	R	?	B

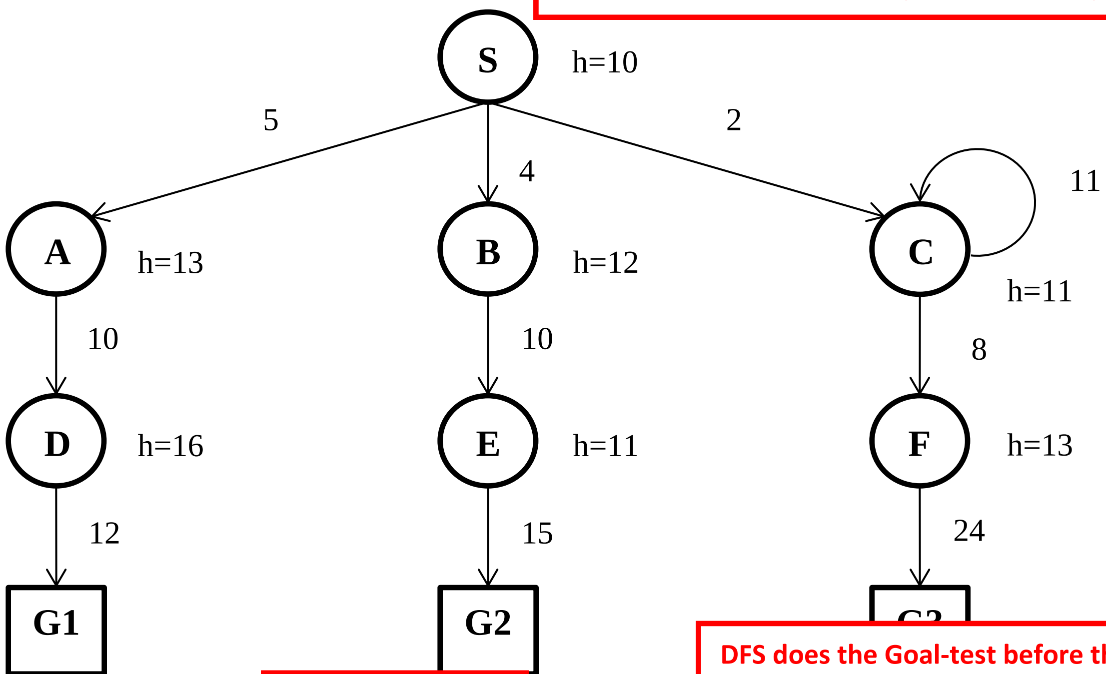
See Section 6.4.

4. (15 pts total, 3 pts each) STATE-SPACE SEARCH. Execute Tree Search through this graph (i.e., 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. 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. Successors are returned in left-to-right order. **(Note: C is a successor of itself).**

For each search strategy below, indicate the or

are generated), ending with the goal node that is found

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.



DFS does the Goal-test before the child is pushed onto the queue. The goal is found when D is expanded.

4.a. DEPTH FIRST SEARCH

See Section 3.4.3 and Fig. 3.17.

S A D G1

4.b. (3 pts, -1 for each wrong a **e) UNIFORM COST SEARCH.**

See Section 3.4.2 and Fig. 3.14.

UCS does goaltest when node is popped off queue.

S C B A F C E D F C G1

4.c. (3 pts, -1 for each wrong answer **REEDY (BEST-FIRST)**

See Section 3.5.1 and Fig. 3.23.

C always has lower h(=11) than any other node on queue.

S C C C C C C C C C C C etc.

4.d. (3 pts, -1 for each wrong answer, **ATED DEEP**

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

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

S S A B C S A D G1

4.e. (3 pts, -1 for each wrong ans **) A* SEARCH.**

See Section 3.5.2 and Figs. 3.24-25.

A* does goaltest when node is popped off queue.

S C B A F C E G2

4.f. (3 pts, -1 for each wrong answer, but not negative) OPTIMALITY.

Did Uniform Cost Search find the optimal goal? Yes Why or why not? Step costs are $\geq \varepsilon > 0$

Did A* Search find the optimal goal? No Why or why not? heuristic is not admissible (at D)

****** TURN PAGE OVER AND CONTINUE ON THI**

OK to say "Heuristic overestimated" or "Heuristic is too high.

5. (15 pts max, -2 for each error, but not negative) MINIMAX WITH ALPHA-BETA PRUNING. Process this game tree left-to-right using Minimax Search with Alpha-Beta pruning.

5.a. What is the best move for MAX? (write A, B, or C) ____________A______

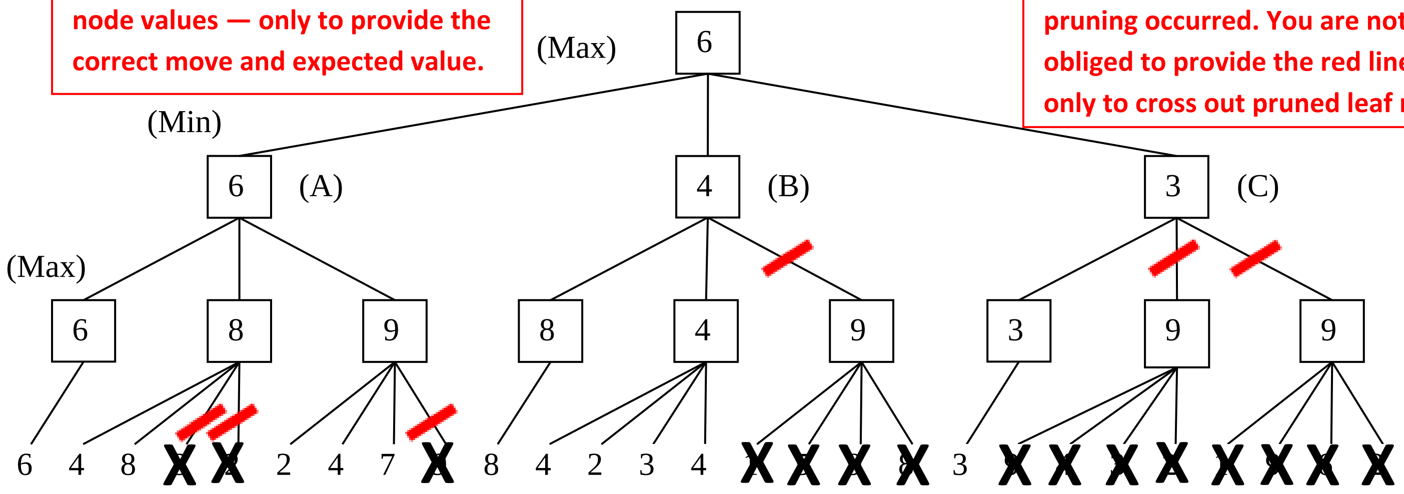
See Section 5.3.

5.b. What value does MAX expect to receive? (write a number) ______________6______

5.c. Cross out each leaf node that will be pruned by Alpha-Beta pruning.

You are not obliged to fill in branch node values — only to provide the correct move and expected value.

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.



6. (10 pts total, -1 each error, but not negative) CONJUNCTIVE NORMAL FORM (CNF). Convert the following logical sentences to Conjunctive Normal Form. **Show your work.**

See Section 7.5.2

6.a. (5 pts total, -1 each error, but not negative). $\mathbf{B \Leftrightarrow (\neg (P \wedge Q))}$

1. Eliminate $\Leftrightarrow$, replacing $\alpha \Leftrightarrow \beta$ with $(\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$.

$(B \Rightarrow (\neg (P \wedge Q))) \wedge ((\neg (P \wedge Q)) \Rightarrow B)$

2. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.

$(\neg B \vee (\neg (P \wedge Q))) \wedge (\neg(\neg (P \wedge Q)) \vee B)$

3. Move $\neg$ inwards using de Morgan's rules.

$(\neg B \vee \neg P \vee \neg Q) \wedge ((P \wedge Q) \vee B)$

4. Apply distributive law ($\wedge$ over $\vee$) and flatten.

$(\neg B \vee \neg P \vee \neg Q) \wedge (P \vee B) \wedge (Q \vee B)$

5. write each clause (disjunct) as a sentence in KB.

$(\neg B \vee \neg P \vee \neg Q)$

$(P \vee B)$

$(Q \vee B)$

6.b. (5 pts total, -1 each error, but not negative). $\mathbf{A \Rightarrow (B \Rightarrow (C \Rightarrow D))}$

1. Eliminate $\Rightarrow$, replacing $\alpha \Rightarrow \beta$ with $\neg\alpha \vee \beta$.

$(\neg A \vee (\neg B \vee (\neg C \vee D)))$

2. Eliminate unnecessary parentheses and write the clause as a sentence in KB.

$(\neg A \vee \neg B \vee \neg C \vee D)$

7. (10 pts total, 2 pts each) RESOLUTION. Apply resolution to each of the following pairs of clauses, then simplify. Write your answer in Conjunctive Normal Form (CNF). If no resolution is possible write "None."

See Section 7.5.2 and Figure 7.13

7.a. (2 pt) (A B ¬C D) (A C D E F). ________(A B D E F)________________________.

7.b. (2 pt) (A B ¬C D) (¬B). ________(A ¬C D)________________________.

7.c. (2 pt) (A B C ¬D) (A C ¬D E F). ________None________________________.

7.d. (2 pt) (¬C) (C). ________( )____**"FALSE" is OK**________________________.

7.e. (2 pt) (A B C ¬D) (A ¬C D E F). ________(A B ¬C C E F) **also OK** (A B D ¬D E F)____.
"TRUE" is OK

8. (15 pts total, -1 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.

See Figure 3.21

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

****** THIS IS THE END OF THE MID-TERM EXAM ******