CS-171, Intro to A.I. — Quiz#3 — Fall Quarter, 2012 — 20 minutes

YOUR NAME: _____

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

1. (35 pts total, -5 pts for each error, but not negative) The Knowledge Engineering process. Your book identifies seven sequential steps in the knowledge engineering process, which steps are below. Unfortunately, the order of the steps has been scrambled. Please, straighten them out.

A. Encode a description of the specific problem instance
B. Assemble the relevant knowledge
C. Pose queries to the inference procedure and get answers
D. Encode general knowledge about the domain
E. Debug the knowledge base
F. Identify the task
G. Decide on a vocabulary of predicates, functions, and constants

See Section 8.4

Fill in the blanks with the letters A, B, C, D, E, F, and G, all in the proper sequence.

F B G D A C E .

2. (30 pts total, 5 pts each) Logic-To-English. For each of the following FOPC sentences on the left, write the letter corresponding to the best English sentence on the right. Use these intended interpretations: (1) "Student(x)" is intended to mean "x is a student." (2) "Quiz(x)" is intended to mean "x is a quiz." (3) "Got100(x, y)" is intended to mean "x got 100 on y."

See Section 8.2.6

B	$\forall s \exists q$ Student(s) $\Rightarrow$ [Quiz(q) $\wedge$ Got100(s, q)]	A	For every quiz, there is a student who got 100 on it.
E	$\exists q \forall s$ Quiz(q) $\wedge$ [Student(s) $\Rightarrow$ Got100(s, q)]	B	For every student, there is a quiz on which that student got 100.
A	$\forall q \exists s$ Quiz(q) $\Rightarrow$ [Student(s) $\wedge$ Got100(s, q)]	C	Every student got 100 on every quiz.
F	$\exists s \forall q$ Student(s) $\wedge$ [Quiz(q) $\Rightarrow$ Got100(s, q)]	D	Some student got 100 on some quiz.
C	$\forall s \forall q$ [Student(s) $\wedge$ Quiz(q)] $\Rightarrow$ Got100(s, q)	E	There is a quiz on which every student got 100.
D	$\exists s \exists q$ Student(s) $\wedge$ Quiz(q) $\wedge$ Got100(s, q)	F	There is a student who got 100 on every quiz.

****** TURN PAGE OVER. QUIZ CONTINUES ON THE REVERSE ******

3. (35 pts total, -5 pts for each error, but not negative) The Horned And Magical Unicorn.

If the unicorn is mythical, then it is immortal, but if it is not mythical, then it is a mortal mammal. If the unicorn is either immortal or a mammal, then it is horned. The unicorn is magical if it is horned. Prove that the unicorn is both horned and magical.

See Section 7.5.2

Use these propositional variables **("immortal" = "not mortal")**:

Y = unicorn is m**Y**thical **R** = unicorn is mo**R**tal **M** = unicorn is a ma**M**mal

H = unicorn is **H**orned **G** = unicorn is ma**G**ical

You have translated your goal sentence, "*horned and magical,*" into (H ^ G), so the negated goal is:

(¬ H ∨ ¬ G)

You have translated the English sentences into a propositional logic Knowledge Base (KB):

(¬ Y ∨ ¬ R) (Y ∨ R) (Y ∨ M)

(R ∨ H) (¬ M ∨ H) (¬ H ∨ G)

Produce a resolution proof, using KB and the negated goal, that the unicorn is horned and magical.

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. Continue until you produce ( ). If you cannot produce ( ), then you have made a more lines, if your proof is correct. The first one is done for you, as an example.

It is OK if you used abbreviated CNF, i.e., (¬ H ¬ G) instead of (¬ H ᵛ ¬ G). It is OK to omit the parentheses.

Resolve (¬ H ∨ ¬ G) and (¬ H ∨ G) to give (¬ H).

Resolve (¬ M ∨ H) and (¬ H) to give (¬ M).

Resolve (Y ∨ M) and (¬ M) to give (Y).

Resolve (¬ Y ∨ ¬ R) and (Y) to give (¬ R).

Resolve (R ∨ H) and (¬ R) to give (H).

Resolve (¬ H) and (H) to give ( ).

Other proofs are OK as long as they are correct.
For example, another proof is:
Resolve (¬ H ᵛ ¬ G) and (¬ H ᵛ G) to give (¬ H).
Resolve (R ᵛ H) and (¬ H) to give (R).
Resolve (¬ Y ᵛ ¬ R) and (R) to give (¬ Y).
Resolve (Y ᵛ M) and (¬ Y) to give (M).
Resolve (¬ M ᵛ H) and (M) to give (H).
Resolve (H) and (¬ H) to give ( ).

A bright and clever student has constructed a shorter proof than I was able to find:
Resolve (¬H ¬G) and (¬H G) to give (¬H)
Resolve (¬Y ¬R) and (Y M) to give (¬R M)
Resolve (¬R M) and (R H) to give (M H)
Resolve (M H) and (¬M H) to give (H)
Resolve (¬H) and (H) to give ()

Resolve _____ and _____ to give _____.

Resolve _____ and _____ to give _____.