CS-171, Intro to A.I. — Quiz#3 — Winter Quarter, 2014 — 20 minutes

YOUR NAME: ____

YOUR ID: ________ ID TO RIGHT:________ ROW:_____ SEAT NO.: ________

1. (35 pts total, -5 pts for each edit step from your answer to the correct answer)
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.

See Section 8.4

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

Due to an ambiguity in answering a student question during the quiz, it is OK for "F" to appear in any position.

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	∀s∃q Student(s) ⇒ [Quiz(q) ∧ Got100(s, q)]	A	For every quiz, there is a student who got 100 on it.
E	∃q ∀s Quiz(q) ∧ [Student(s) ⇒ Got100(s, q)]	B	For every student, there is a quiz on which that student got 100.
A	∀q∃s Quiz(q) ⇒ [Student(s) ∧ Got100(s, q)]	C	Every student got 100 on every quiz.
F	∃s ∀q Student(s) ∧ [Quiz(q) ⇒ Got100(s, q)]	D	Some student got 100 on some quiz.
C	∀s∀q [Student(s) ∧ Quiz(q)] ⇒ Got100(s, q)	E	There is a quiz on which every student got 100.
D	∃s∃q Student(s) ∧ Quiz(q) ∧ 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 edit step from your proof to a correct proof)
Resolution Proof. (http://www.braingle.com)
Detective Dorothy interviewed four local burglars to identify who stole Lady Diva's teapot.

See Section 7.5.2

<u>It was well known that each burglar told exactly one lie:</u>

> ***Arnold:*** *I didn't do it. Brian did it.*
> ***Brian:*** *I didn't do it. Derek did it.*
> ***Charlie:*** *I didn't do it. Brian is lying when he says Derek did it.*
> ***Derek:*** *I didn't do it. If Arnold didn't do it, then Brian did it.*

Use these propositional variables:

> **A**= Arnold did it. **B**= Brian did it. **C**= Charlie did it. **D**= Derek did it.

You translate the evidence into propositional logic (<u>recall that each suspect told exactly one lie</u>):

> ***Arnold:*** $(A \wedge B) \vee (\neg A \wedge \neg B)$ ***Brian:*** $(B \wedge D) \vee (\neg B \wedge \neg D)$
> ***Charlie:*** $(C \wedge \neg D) \vee (\neg C \wedge D)$ ***Derek:*** $(D \wedge (\neg A \Rightarrow B)) \vee (\neg D \wedge \neg (\neg A \Rightarrow B))$

At most one burglar stole the teapot:

> $(A \Rightarrow \neg B \wedge \neg C \wedge \neg D)$ $(B \Rightarrow \neg A \wedge \neg C \wedge \neg D)$ $(C \Rightarrow \neg A \wedge \neg B \wedge \neg D)$ $(D \Rightarrow \neg A \wedge \neg B \wedge \neg C)$

After converting to Conjunctive Normal Form, your Knowledge Base (KB) consists of:
Arnold: $(A \vee \neg B)$ $(\neg A \vee B)$ ***Brian:*** $(B \vee \neg D)$ $(\neg B \vee D)$
Charlie: $(C \vee D)$ $(\neg C \vee \neg D)$ ***Derek:*** $(\neg A \vee D)$ $(\neg B \vee D)$ $(A \vee B \vee \neg D)$
At most one: $(\neg A \vee \neg B)$ $(\neg A \vee \neg C)$ $(\neg A \vee \neg D)$ $(\neg B \vee \neg C)$ $(\neg B \vee \neg D)$ $(\neg C \vee \neg D)$
(Side note: Normally, you would start four proofs, one for each goal sentence: A, B, C, D. Only the proof of C would succeed, and you would know Charlie did it. For this timed test, you will do only one proof.)
You will be asked to prove, **"Charlie did it."** The goal is (C). You adjoin the negated goal to your KB:

> $(\neg C)$

Other proofs are OK as long as they are correct. For example, a three-line proof is:
Resolve (B ˅ ¬D) and (¬B ˅ ¬D) to give (¬ D).
Resolve (C ˅ D) and (¬ D) to give (C).
Resolve (C) and (¬ C) to give ( ).

Produce a resolution p
Repeatedly choose two
the second. Apply resol
the knowledge base. Continue until you produce ( ). If you cannot produce ( ), then you have made a mistake. The shortest proof I know is only three lines. It is OK to use more lines, if your proof is correct. It is OK to use abbreviated CNF, i.e., (¬A ¬B) instead of (¬A ∨ ¬B). It is OK to omit the parentheses.

Resolve $(C \vee D)$ and $(\neg C)$ to give (D).

Resolve $(B \vee \neg D)$ and (D) to give (B).

Resolve $(A \vee \neg B)$ and (B) to give (A).

Resolve $(\neg A \vee \neg B)$ and (B) to give $(\neg A)$.

Resolve $(\neg A)$ and (A) to give $( )$.

Resolve ________________

Resolve ________________

Other proofs are OK as long as they are correct. For example, another proof is:
Resolve (¬ C) and (C ˅ D) to give (D).
Resolve (D) and (¬ B ˅ ¬ D) to give (¬ B).
Resolve (¬ B) and (B ˅ ¬ D) to give (¬ D).
Resolve (¬ D) and (D) to give ( ).

Extra lines or steps are OK as long as your proof is correct.

It is OK (even expected) to simplify expressions. E.g., if you resolved (¬ B ˅ ¬ D) and (B ˅ ¬ D) to give (¬ D ˅ ¬ D), of course you would simplify it to (¬ D). It is OK to simplify as you go, i.e., you don't need a separate step.