**CMSC 471 HW4 Spring 2022**

**Your Name (XYZZY@umbc.edu)**

Due April 12; total points: 100

Edit this file to include your answers either in Word or Google Docs. We done some of the for you as examples. Save both a hw4.docx and hw4.pdf version of the file with answers in your repository and commit and push them back to your GitHub repo.

**1. Checking Validity (10 points)**

1.a P ∨ ¬P

>>> tt\_true(expr('P | ~P'))  
True

1.b (P → Q) ↔ (¬P ∨ Q)

Your answer here

1.c P ∧ Q → (Q ∨ P)

Your answer here

1.d (P ∧ Q) → Q

Your answer here

1.e ((A ∧ B) → C) ↔ (A → (B → C))

Your answer here

1.f ((a → b) → a) → a

Your answer here

**2. Satisfiability (14 points)**

*2.a P ∧ Q*

*>>> dpll\_satisfiable(expr('P & Q'))  
{P: True, Q: True}*

2.b ALIVE ↔ ¬DEAD

Your answer here

2.c P → ¬ P v P

Your answer here

2.d ¬ (P ∨ ¬ P)

Your answer here

2.e P ∧ (P → Q)

Your answer here \

2.f P ∧ (Q →P)

Your answer here

2.g P ∧ (Q ∨ ¬P)

Your answer here

2.h P ∧ ¬Q ∧ (P → Q)

Your answer here

**3. Propositional Consequence (14 points)**

For each of the following entailment relations, say whether or not it is true. The text on the left of the entailment symbol (⊨) represents one or more sentences (separated by commas) that constitute a knowledge base. We've done the first one for you.

*3.a P ∧ Q ⊨ P*

*True*

3.b P ⊨ P ∨ Q

Your answer here

3.c ¬P ⊨ ¬ ¬ P

Your answer here

3.d P → Q ⊨ ¬ P → ¬ Q

Your answer here

3.e ¬ P ⊨ P → Q

Your answer here

3.f ¬ Q ⊨ P → Q

Your answer here

3.g P ∧ (P → Q) ⊨ Q

Your answer here

3.h ( ¬ P) ∧ (Q → P) ⊨ ¬ Q

Your answer here

**4. English to FOL (25 points)**

Translate the following English sentences into first order logic, describing the intended meaning of any non-obvious predicates. Feel free to optionally provide a more direct *paraphrase* of the meaning of your logic expression in English. If you think a sentence is ambiguous, describe the ambiguity and give logical expressions for all interpretations. We've done the first one for you using a notation with simple ASCII characters for logical operators (e.g., A:∀, E:∃, =>:→, <=>:↔, ^:∧, v:∨, ~:¬, >:>)

*4.1 There is no largest prime number.*

*~(Ex number(x)^prime(x)^(Ay number(y)^ prime(y) -> y >= x))*

***paraphrase:*** *It is not true that there is a prime number and it is greater than or equal to all prime numbers.* ***predicates:*** *number(x) is true if and only if x is a positive integer and prime(x) is true if and only if x is a prime number.*

4.2 Good food is not cheap and cheap food is not good.

Your answer here

4.3 John has exactly two brothers.

Your answer here

4.4 Every dog is either male or female and no dog can be both male and female.

Your answer here

4.5 The friend of your enemy is your enemy.

Your answer here

4.6 An ancestor of your ancestor is your ancestor.

Your answer here

**5. Expressing knowledge in CNF (15)**

Express each of the above as a set of clauses in conjunctive normal form (CNF)

*a. A ∨ (B ∧ C)*

*A ∨ B*

*A ∨ C*

b. A ∨ B ∨ C

Your answer here

c. A ∧ B ∧ C

Your answer here

d. (A ∧ B) ⇔ C

Your answer here

e. A => (B ⇔ C)

Your answer here

f. A => (( B => C) ∨ ~C)

Your answer here

**6. Solving a puzzle with SAT (22 points)**

There are three robots: WALL-E, R2D2 and 3CPO. You want some help writing software and think that one of them can do it. You know that

* One and only one of them knows Python
* One of the three robots always tells the truth and the other two always lie

but you don't know which one knows Python and which one is the truthful one. So, you ask each one which of the three knows Python, and they give the following answers.

* WALL-E: “I know Python”
* R2D2: “I do not know Python”
* C3PO: “WALL-E does not know Python”

What the bots don't know is that you are a smart person rather than a robot and can use logic to reveal who is lying and who is telling the truth and discover which one of them knows Python. Explain your reasoning by (a) mapping the problem into propositional logic and (b) showing how the AIMA code can be used to solve this problem.

Hint: These puzzles are hard for people because they are self-referential. Here's a suggestion.

* Start by trying to figure out the puzzle though a process of elimination
* Create variables to represent who knows Python (e.g. P1 might be true if *WALL-E knows Python*)
* Create sentences for each of the three statements (e.g., S1 is what WALL-E says: *WALL-E knows Python*)
* Create sentences that capture the constraints that only one knows Python and only one is telling the truth
* Use **dpll\_statisfiable** on a conjunction of these local sentences to see what model satisfies them

If you've done things correctly, the sentences will be satisfiable and the model will reveal who know Python. Show your logical sentences and the model they produce.

Your answer here