CSCI 5511 Final

Due Monday, Dec 21, 11:59 pm

- This is a *take-home* exam. It is open book, open web, but closed people. You may not consult with anyone other than Andy regarding the questions and information on this exam.
- Show all work, clearly and in order, if you want to get full credit. I reserve the right to take off points if I cannot see how you arrived at your answer (even if your final answer is correct).
- If you think a question is unclear or ambiguous, or if you think that there is an error on the exam, either state your assumptions or contact Andy.
- One of the questions has a code component that should be submitted via Canvas. Please comment your code explaining how to run it.
- Your other answers should be submitted as a pdf via Canvas.
- This test has 6 problems and is worth 56 points.
- Good luck!

1. Propositional Logic (*6 points*) For each of the following assertions in propositional logic, either prove that it is true, or else provide a counterexample:

a. (*2 pts*) If $\alpha \models \gamma$ or $\beta \models \gamma$ (or both), then $\alpha \wedge \beta \models \gamma$.

b. (*2 pts*) If $\alpha \models (\beta \vee \gamma)$ then $\alpha \models \beta$ or $\alpha \models \gamma$.

c. (*2 pts*) If $\alpha \models (\beta \wedge \gamma)$ then $\alpha \models \beta$ and $\alpha \models \gamma$.

2. Propositional Logic (*12 points*) Consider a sentence in propositional logic in Conjunctive Normal Form (CNF) with exactly k symbols in each group of conjunctions. Such a sentence is defined to be in k-CNF form. For example, a possible sentence in 3-CNF form for a world with 5 symbols (A, B, C, D, E) and 4 clauses is:

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

Given a random k-CNF sentence, it may or may not be satisfiable.

For this problem, write a program named `randompl.py`.

a. (*4 pts*) Write a function `rand3cnf` that takes m and n as parameters, then generates a random sentence in 3-CNF form with m clauses and n symbols. For example, calling this function with $m = 4$ and $n = 5$ might generate the sentence that was given as an example above.

For each clause, you should make certain that literals are not repeated (i.e. you should be able to generate the clauses above, but not a clause of the form $A \vee A \vee B$ or $A \vee B \vee \neg A$)

b. (*4 pts*) Write a function that will call the function from part (a) for m from 20 to 80 (in steps of 10) and n in $[5, 10, 15, 20]$. For each (m, n) setting, generate 100 random knowledge bases and use `SATInterface` to test whether they are satisfiable or not. Have this program print out the results for each.

c. (*4 pts*) For what relationships between m and n is the probability of satisfiability "interesting" (i.e. not 0 or 1)? Describe why you got your answer based on your results from part (b)

3. First-Order Logic (*6 points*) Let us consider using First-Order Logic to represent a mathematical domain. Our world will have constant symbols 0 and 1, the binary predicate $<$ and the function symbols $+$ and $\times$.

Additional predicates can be defined using biconditionals.

a. (*2 pts*) Represent the property "x is an odd number".

b. (*2 pts*) Represent the property "x is prime".

c. (*2 pts*) There is a mathematical conjecture that every odd integer greater than 7 can be written as the sum of three odd primes. Write this conjecture as a logical sentence.

4. First-Order Logic (*10 points*) Consider the following two sentences in first order logic:

α $\forall x \; \exists y \; (x \geq y)$

β $\exists y \; \forall x \; (x \geq y)$

a. (*2 pts*) Assume that the variables range over the domain of natural numbers $(0, 1, 2, ...)$ and that the predicate $\geq$ means "greater than or equal to." Under this interpretation, translate α and β into English sentences.

b. (*2 pts*) Under your translation, are these sentences true?

c. (*2 pts*) Does $\alpha \models \beta$, or does $\beta \models \alpha$, both, or neither?

d. (*4 pts*) Prove your findings from part C. (All of your findings, that is, prove that things are true and also prove that things are not true.)

5. Planning (*10 points*) Consider the following problem:
An agent is trapped in a puzzle room! To escape from the main room into the hallway, the agent must place boxes onto switches on the floor.

- There is initially only one box in the room.
- There are two doors, A and B, leading out of the room.
- There are three switches on the floor.
- Placing a box on Switch #1 will open door A, which leads to the closet.
- Placing a box Switch #2 and another box on Switch #3 will open door B, which leads to the hallway.
- Inside the closet is a second box.
- If you remove the boxes from the switches, then the doors close again.

Using the following predicates

- In(x, y)
- Room(x)
- On(x, y)
- Block(x)
- Door(x)
- Open(x)

and any other predicates that you deem necessary, answer the following questions:
(You may also want to add and define some constants that represent objects)

a. (*2 pts*) What is the initial state for this planning problem?

b. (*2 pts*) What is the goal state for this planning problem?

c. (*6 pts*) What are the PDDL action schema that are required for an agent to use planning to find a solution?

6. Problem Approaches (*12 points*) For each of the following puzzles, consider what sort of formulation of the problem would work from the following:

- Uninformed Search
- Informed Search
- Local Search
- Constraint Satisfaction
- Planning

Describe two approaches for each problem and explain (using algorithm comparisons) why one is better than the other.
Part of your description should involve how you plan to represent the world of the problem.

a. (*4 pts*) Knight's tour - Find a way for a knight to visit all 64 squares on a chessboard without visiting the same space twice.

b. (*4 pts*) Crossword puzzle - Figure out how to put english words into a grid-like structure that forms words going across and down. (For those who have encountered crossword puzzles before, you can ignore the clues part of this and just consider putting legal words into the grid as your goal.)

c. (*4 pts*) Roomba™ source code - Having a robotic vacuum cleaner visit all locations in aroom