Sample Midterm 2
CSCI 561 Fall 2023: Foundation of Artificial Intelligence

Instructions:

1. Maximum credits/points for this midterm: 100 points.
2. No books (or any other material) are allowed.
3. Be brief: a few words are often enough if they are precise and use the correct vocabulary studied in class.s
4. Adhere to the Academic Integrity Code.
5. **Add suggested symbol usage**

Problems	100 Percent Total
1 – True/False	10%
2 – Propositional Logic	10%
3 – First Order Logic	20%
4 – Inference	20%
5 – CNF Transformation (skolemization)	10%
6 – Planning	20%
7 – Multiple Choice	10%

1. True/False [10%]

For each of the statements below, fill in the bubble **T** if the statement is always and unconditionally true, or fill in the bubble **F** if it is always false, sometimes false, or just does not make sense:

1. Hill climbing is a complete search algorithm for global solutions and it never looks ahead beyond the immediate neighbors of the current state.
2. “The sentence A entails the sentence B” if and only if, in every model in which B is true, A is also true.
3. An inference algorithm f is complete means that if A can inference B using f, then A entails B.
4. An inference algorithm f is sound means that if A entails B, then A can inference B using f.
5. A sentence is valid if it is true in all models.
6. “From Hates(Alice, Exam) we can infer ∃ x Hates(x, Exam)” applies the Existential Ξlimination (EE) rule in First-order Logic.
7. Logical agents apply entailment to a knowledge base to derive new information and make decisions.
8. Ξvery propositional formula can be converted into an equivalent formula that is in Conjunctive Normal Form.
9. After applying Skolemization to a sentence, the resulting formula is satisfiable if and only if the original sentence is satisfiable.
10. Horn clause is a disjunction of literals of which at least one is positive.

2. Propositional Logic [10%]

1. Translate the following English sentences to Propositional Logic.
   Let:
   - E=Liron is eating
   - H=Liron is hungry

   Choose from the following options
   a) $E \Rightarrow \neg H$
   b) $\neg(H \Rightarrow \neg E)$
   c) $E \wedge \neg H$
   d) $E \Leftrightarrow H$

   (1) If Liron is eating, then Liron is not hungry.
   (2) Liron is eating and not hungry.
   (3) Liron is hungry and eating.
   (4) Liron is eating if and only if Liron is hungry.

2. Are the following statements true or false? Type T for true and F for false.
    a. TRUE $\models$ FALSE
    b. $(P \wedge Q) \models (P \Leftrightarrow Q)$
    c. $(P \Leftrightarrow Q) \wedge (\neg P \vee Q)$ is satisfiable.
    d. $((P \vee Q) \Leftrightarrow P) \vee (\neg P \wedge Q)$ is a tautology.
    e. From "$P \Rightarrow Q$" and "P", we can infer "Q" by applying the Unit Resolution Rule.
    f. $(P \wedge Q) \Rightarrow R$ is a Horn clause.

3. First Order Logic [20%]

3.A

USC Innovation Sciences Institute (ISI) is currently working on a cool project but the project leads feel like different student workers ask the same questions repeatedly and it is very time-consuming to explain each and everything everytime. One of the leads asked Professor Shen's teaching team's help in making a documentation of some logics. Professor Shen advised us to give such easy tasks to students in the exam-2.

However, there is a prerequisite that you all must know and following is the necessary information that will help you complete the task.

First Order Logic for Graphs:

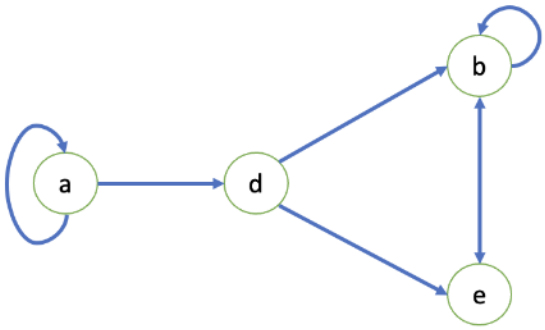
- **The first–order language of (directed) graphs is L = {r}, where r is a binary relation symbol.**
- **The only terms are the variables 'x', 'y', 'z' etc.**
- **Atomic formulas look like (rxy).**
- **Universal and Existential quantifiers work the same way as they normally do.**

Sometimes, smart people have a hard time understanding the above points, and since, you all are smart people, we want to explain the above points in simple terms:

- **For nodes/vertices/terms x and y in a (directed) graph G, rxy denotes that there is an edge going from node x to node y in G.**

Please note that all of the above points are extremely important for you to understand.

To help you understand better, here are a few examples for the following graph G.



Following statement is **false** for the above graph:

$\forall$ **x (rxx)** says "The graph is reflexive" meaning, for every node x, there is an edge from x to x.

Following is **true** for the above graph:

$\exists$ **y (ray)** says "there is an edge going from a to some vertex y"

You have to write the following sentences in first order logic for graphs(Directed). ***Don't think about the graph given as an example for the given questions.***

NOTE: Please follow the same format, as in, **rxy** without any spaces.

1. The graph has at least two vertices. [3%]

2. Every vertex has an edge attached to it. [3.5%]

3. Every vertex has at most two edges directed from it to other vertices. [4.5%]

4. Every vertex has at least two edges directed from other vertices to it. [2%]

5. Write T for True, F for False: [2%]
   "Every vertex has exactly one edge entering it" can be written as:
   $\forall x \exists y\ rxy \wedge \forall w \forall v \forall u\ ((rvw \wedge ruw) \Rightarrow (v = u))$

3.B
Consider a domain with the following relation.

ISPRIME(n) : n is a prime number
HasSS#(x,n) : x has the social security number n
Person(x): x is a person

NOTE:
In a case like, "Every Positive Integer .. or Every Natural Number.. ", you can safely assume that it is the set of integers/numbers that we are talking about, you don't need the predicates Number(x) or Integer(x) in those cases. However, the property "positive" integer or "natural" numbers have to be handled/managed using FOL.

1. Write T for True, F for False: [1%]
   "Every even natural number n has 2 as it's factor" can be written as:
   $\forall$ n. ((n > 2 $\wedge$ $\exists$ k. n = 2k)

2. Goldbach's Conjecture says that "Every even natural number n > 2 can be expressed as the sum of two primes."
   Write Goldbach's Conjecture in First Order Logic. [2%]

3. Write T for True, F for False: [2%]
   "No two people have the same social security number." can be written as:
   $\neg \exists$ x,y,n Person(x) $\wedge$ Person(y) $\wedge$ HasSS#(x,n) $\wedge$ HasSS#(y,n) .

4. Inference [20%]

A. [8%] We designed an exam, which has three parts (P1, P2 and P3) for two types of examinees (E and non-E). For each part, there are three ranks (A, B and C). The result for the entire exam is either pass or not pass.
Given the following sentences:

1. [2%] Any E examinee who gets an A in both part P1 and P2 will pass the exam.
2. [2%] Any non-E examinee who gets a C in part P3 or P2 won't pass the exam.
3. [2%] Some non-E examinees who get a C or B in part P2 and pass the exam.
4. [2%] Alex is an E examinee getting an A in part P1 and Steve is a non-E examinee getting an A in part P3.

Translate the above sentences into First Order Logic using the following predicates: **E(x)** which means x is an E examinee **R(x, y, z)** which means examinee x gets a y in part z, and **P(x)** which means x passes the exam.

Note: sort your clauses primarily in alphabetical order and secondarily in numerical order wherever sortable. For example E(x) should be before R(x, y, z), and R(x, B, P1) should be before R(x, B, P2) but after R(x, A, P2).

1.
2.
3.
4.

B. [12%] Given the following knowledge base where w, x, y, z are variables, P, Q, R, S, T, U are predicates and A, B, C are constants:

1. ¬T(z) ∨ ¬P(A, B, z)
2. ¬Q(A, x) ∨ R(x, C)
3. ¬T(y) ∨ ¬S(A, y)
4. ¬U(A, z) ∨ T(z)
5. U(w, C) ∨ P(w, A, C)
6. ¬P(A, A, C)
7. S(A, y) ∨ P(A, y, C)
8. P(A, x, C) ∨ Q(A, x)
9. U(w, B) ∨ P(w, A, C)
10. P(A, B, B)

Prove the following sentence **using resolution (you can't get points by using other approaches):**

R(B, C)

Note: Sort your reference numbers from small to large, and write N/A if there are no references.
Your proof should be in the following format:

[number, starting from 11] new sentence [N/A, N/A]

[number] new sentence [reference number 1, reference number 2]

…

[number] new sentence [reference number 1, reference number 2]

Contradiction with line [number]

For example:

[11] ~R(B,C)

[N/A,N/A] (negation of the query)

[12] ~Q(A,B) [2,11]

5. CNF Transformation (skolemization) [10%]

Convert the following sentence into Conjunctive Normal Form (CNF):

∀x [(∀y (N(y) => L(x,y))) => ~(P(x) => ∀y (L(y,x)))]

Fill in the blanks:

1. The two Skolem Functions being used are F(.) and G(.)
2. No whitespaces
3. No unnecessary brackets
4. Use the character "~" for "NOT"
5. Uppercase letters for functions, lowercase letters for variables

The following is the sentence obtained after performing all except the last step of the CNF transformation (right before the final step of converting to conjunctions of disjunctions):

(N(F(x)) ^ ____1____) v (____2____ ^ ~____3____)
____1____:
____2____:
____3____:

Denoting N(F(x)) as "4", and denoting your answers above by the blank number they fill (i.e. your answer for ____1____ will be denoted as "1"), which of the following is the final CNF form of the given sentence?:

a) (4 v 2) ^ (4 v 1) ^ (1 v ~3) ^ (2 v ~3)
b) (4 v 2) ^ (4 v ~3) ^ (1 v 2) ^ (1 v ~3)
c) (4 v 2) ^ (4 v ~3) ^ (1 v ~3) ^ (2 v ~3)
d) (4 v 2) ^ (4 v 1) ^ (1 v 2) ^ (1 v ~3)

6. Planning [20%]

Stack is one of the basic data structures in programming. The following line is a stack, with 3 as its top element and 7 as its bottom:

Stack S: Head→3→7→5→4

For a stack, we have following basic actions:
Pop(s): pop the top of stack s.
Push(s, e): push element e to the top of stack s.

And the following conditions:

IsHead(s, e1): e1 is the head of stack s. For example, in the given stack, we have IsHead(S, 3).
IsEmpty(s): stack s is empty.
NextTo(e1, e2): element e1 is the element before e2. For example, in the given stack, we have NextTo(3,7).

Note: In your answers, use "e1, e2, e3 …" for non-constant elements, use variable "s" for non-constant stack.

A. [4%] What are the current conditions for the given stack? List **all** of them.

Write your answer in the following format:

Condition1(variable1, variable2, …)
Condition2(variable1, variable2, …)
…

B. [6%] What are the preconditions and postconditions for action Pop(s)? For pop(), you remove the top element in the sequence. (Assume the stack is not empty for this question.)

Write your answer in the following format:

Situation 1:

Preconditions:
Condition1(variable1, variable2, …)
Condition2(variable1, variable2, …)
…

Postconditions:
Condition1(variable1, variable2, …)
Condition2(variable1, variable2, …)
…

Situation 2: (if there are multiple situations)
…

C. [7%]In order to sort the elements in the stack **(the top element is the smallest)**, we add a new action:
Switch(e1, e2): switch element e1 and e2 when NextTo(e1, e2), only when e1 is larger than e2.
and a new condition:
Larger(e1, e2): element e1 is larger than e2 when NextTo(e1, e2).

What are the preconditions and postconditions for Switch(e3, e4) in a subsequence e1→e2→e3→e4→e5?

Write your answer in the same format as question B:

D. [3%]Provide a possible solution (write down the sequence of actions) to sort the given stack **S: Head→3→7→5→4**.
(the top element is the smallest):

7. Multiple choice [10%]

1. In the discussions, we showed the definitions of entailment, please select all that are true:
   a. $KB \models \alpha$ if and only if $KB \rightarrow \alpha$ is satisfiable.
   b. $KB \models \alpha$ if and only if $KB \rightarrow \alpha$ is valid.
   c. $KB \models \alpha$ if and only if $KB \wedge \neg\alpha$ is unsatisfiable.
   d. $KB \models \alpha$ if and only if $KB \wedge \neg\alpha$ is satisfiable.
   e. $KB \models \alpha$ if and only if $\neg KB \vee \alpha$ is valid.

2. In the discussions, we compared several logical inference methods.
Let $KB \equiv \{A \rightarrow B, \neg A \rightarrow C, B \rightarrow D, C \rightarrow D\}$, please circle all that are true
   a. It is possible to prove D from KB using Backward chaining
   b. It is impossible to prove D from KB using Forward chaining
   c. It is possible to prove D from KB using Resolution
   d. It is impossible to prove D from KB at all
   e. It does not make sense to prove D from KB

3. If C is a constant, x is a variable, and F is a function, which of these will not unify?
   a. x and F(x)
   b. x and C
   c. C and F(x)
   d. F(C) and F(x)

4. Which one of the following is not true of ground literals?
   a. They have only universally quantified variables as arguments.
   b. They behave just like propositional symbols in automated reasoning.
   c. They might unify with literals containing only universally quantified variables.
   d. They may contain terms that are functions.

5. What is the English equivalent of "$\neg \exists x\ (Human(x) \wedge Perfect(x))$"?
   a. Not every human is perfect.
   b. There is no Perfect human.
   c. Being a human, implies not being perfect.
   d. Some humans are perfect.