Midterm 2
CSCI 561 Fall 2022: 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	15%
3 – First Order Logic	15%
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. $(A \Leftrightarrow B) \wedge (\neg A \vee B)$ is valid. F
2. First Order Logic has quantifiers $\forall$ and $\exists$. T
3. In FOL, constant symbols refer to relations, while predicate symbols refer to objects. F
4. Sound inference algorithms are always complete. F
5. “Everything attracts something”, where “something” means “something or other”, is equivalent to “$\forall x\ \forall y\ A(x, y)$” [Given that Attract is a relation from x to y, i.e., A(x,y) says that “x attracts y” or equivalently that “y is attracted by x”.] F
6. Linearization is the process of deriving a totally ordered plan from a partially ordered plan. T
7. Skolemization is the process of removing universal quantifiers by elimination. F
8. All sentences can be expressed in Horn form. F
9. The completeness theorem says that a sentence can be proved if it is entailed by another set of sentences. T
10. First Order Logic is monotonic. T

2. Propositional Logic [15%]

Consider the following KB and α:

$KB = (p \rightarrow \neg q) \wedge (r \rightarrow q) \wedge (\neg r \rightarrow p)$

$\alpha = ((\neg p \wedge q) \vee (p \wedge \neg q)) \wedge ((q \wedge r) \vee (\neg q \wedge \neg r)) \wedge (p \vee \neg q)$

Please fill in the truth table with "T" or "F" and answer the following questions [8%]

(This section will be graded automatically)

p	q	r	KB	α
F	F	F	F	F
F	F	T	F	F
F	T	F	F	F
F	T	T	T	F
T	F	F	T	T
T	F	T	F	F
T	T	F	F	F
T	T	T	F	F

Manual Grading (first)

19 (a) Does $KB \vDash \alpha$, why or why not? [3%]

No, there are some cases that KB is true but α is not.

20 (b) Is KB satisfiable? [1%]

Yes

21 (c) Is α satisfiable? [1%]

Yes

22 (d) Is KB Valid? [1%]

No

23 (e) Is α Valid? [1%]

No

3. First Order Logic [15%]

Consider a domain with the following relations and objects.

Eats(x,y)	-	Person x eats Food y
Tastes(x,y)	-	Person x tastes Food y
Cooks(x,y)	-	Person x cooks Food y
Person(x)	-	x is a Person
Customer(x,y)	-	Person x is a customer of Person y
Chef(x)	-	Person x is a chef
Food(y)	-	y is Food.
LivesAlone(x)	-	Person lives alone
Meat, Vegetables, Fruit	-	Constants denoting Food

Formalize the following sentences for this domain.

Manual Grading

1. **24** [3%] There is no Chef who doesn't taste all of the food they cook.

$\neg \exists x \forall y$ Food(y) $\wedge$ Chef(x) $\wedge$ Cooks(x,y) $\wedge$ $\neg$Tastes(x,y)

$\forall x \forall y$: Chef(x) ^ Food(y) ^ Cooks(x,y) => Tastes(x,y)

2. **25** [5%] There is a chef who cooks meat, but is not a customer of any chef that cooks meat

$\exists y$ { Chef(y) $\wedge$ Cooks(y, Meat) $\wedge$ $\forall x$[Chef(x) $\wedge$ Cooks(x,Meat) => $\neg$Customer(y, x)] }.

3. **26** [4%] Any person who does not cook any food either does not live alone or is a customer of at least one chef.

$\forall x \forall y$ (Person(x) $\wedge$ Food(y) $\wedge$ $\neg$Cooks(x, y)) => ($\neg$LivesAlone(x) $\vee$ $\exists z$ (Chef(z) ^ Customer(x,z)))

4. **27** [3%] Every chef who eats food is a customer of a chef.

$\forall x \exists y \exists z$ Chef(x) $\wedge$ Food(y) $\wedge$ Eats(x, y) => Chef(z) $\wedge$ Customer(x, z)

4. Inference [20%]

1. **28** Manual Grading: (12 Points)

Prove KB |= α using contradiction. KB and α are defined as follows:

KB: (p $\rightarrow$ q), ($\neg$r v s), (p v r)
α: ($\neg$q $\rightarrow$ s)

Fill the rest of the table to complete the proof:

Resolvent	**Sentence1, Sentence2, …., Rule used**
1. (p $\rightarrow$ q)	Premise
2. $\neg$($\neg$q $\rightarrow$ s)	Adding $\neg\alpha$ to the KB
3. $\neg$q $\wedge$ $\neg$s	S2, Simplifying the implication and distributing $\neg$

Please use the above format for your answer (left side resolvent, right side justification).

(0 Points if not proved using contradiction)
Partial - (1 Point for each correct resolvent if it is leading to the right solution, 0.5 if reason provided for a step is correct)

Solution:

4. ($\neg$r v s)	Premise
5. (p v r)	Premise
6. $\neg$s	S3, Conjunctive Simplification
7. $\neg$r	S4, S6, Disjunctive Syllogism
8. $\neg$q	S3, Conjunctive Simplification
9. $\neg$p	S1, S8, Modus Tollens
10. r	S5, S9, Disjunctive syllogism
11. $\neg$r $\wedge$ r	S7, S10, And Introduction

Since S11 is a contradiction, S2 can't be true

2. If “x = 10”, then “there is no solution”. “There is no solution”, therefore “x =10”. Is the above inference correct or not? (3 Points)

Solution: No

Explanation: If ‘x = 10’ is p and ‘there is no solution’ is q. The argument is translated to logic as the inference $\{p \rightarrow q, q\} \Rightarrow p$? We need to determine whether $(p\rightarrow q)\wedge q\rightarrow p$ is a tautology or not.

$(p\rightarrow q) \wedge q$ (And Introduction)

$= (\neg p \vee q) \wedge q$ (Simplifying the implication)

$= (\sim p \wedge q) \vee (q \wedge q)$ (Distributing $\wedge$)

$= (\sim p \wedge q)$ Step 4

$= \sim p$

$= F$

1. $p \rightarrow q$ Premise
2. q Premise
3. $\neg p \vee q$ Simplifying the implication

We can not infer if p is T/F from this KB therefore we can not prove $\{p \rightarrow q, q\} \Rightarrow p$

Hence the argument is invalid

3. If there is no solution, x = 10. There is no solution, therefore x = 10.

Is the above argument a valid one? (2 Points)

Solution: Yes

Explanation: It translates to $\{p \rightarrow q, p\} \Rightarrow q$? Modus Ponens

4. If $p\rightarrow q$ and $p\rightarrow r$, can we conclude that $p\rightarrow(q\wedge r)$? (3 Points)

Solution: Yes

Explanation:

1. $p \rightarrow q$ Premise
2. $p \rightarrow r$ Premise
3. $(p \rightarrow q) \wedge (p \rightarrow r)$ S1, S2, And Introduction
4. $p \rightarrow (q\wedge r)$ S3, Factoring

5. CNF Transformation (skolemization) [10%]

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

$\forall x[\ \forall y\ A(y) \rightarrow L(x,y)\ ]\ \rightarrow [\ \exists y\ L(y, x)]$

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

Q1) **32** 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):

Manual Grading (last one)

(A(F(x)) ^ **~** ____1____) v (____2____)

____1____:

____2____:

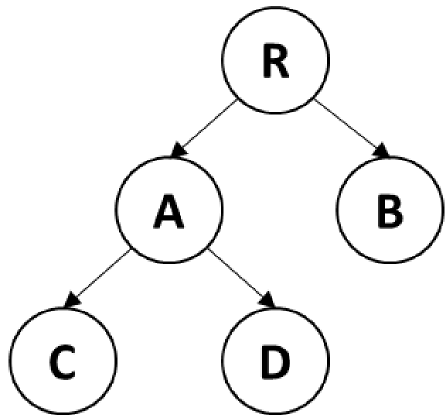
1. L(x,F(x)) [4%]
2. L(G(x),x) [4%]

Q2) Denoting A(F(x)) as "3", 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) (3 v 2) ^ (~1 v 2) [2%]
b) (3 v 2) ^ (1 v 2)
c) (3 v ~2) ^ (1 v ~2)
d) (~3 v 2) ^ (1 v 2)

6. Planning [20%]

Tree, one of the basic data structures in computer science, describes hierarchical relations between entities. The figure below depicts a tree:



In the given sample tree, R is the root, A and B being R's children and C, D being A's children.

We now define two valid actions for a tree:

- **addChild(X, Y)**: Let Y be a child of X. We will have X->Y in the tree.
- **removeNode(X)**: Remove node X from the tree. When it has children, its children will become children of its parent node. If X is the root, simply delete the entire tree.

And the following conditions:

- **isRoot(X)**: Some node X is the root of the given tree. For example, in the sample tree, we have isRoot(R).
- **isEmpty()**: The given tree is empty, which means there is no node in the tree.
- **pointTo(Y, X)**: Some node X points to some node Y, which means X is the parent node of Y. For example, in the given stack, we have pointTo(A,R), etc.

Note:

- The names of all entities, conditions and actions in this question are **case-sensitive**.
- For pre and post conditions, you should **only** include conditions that are impacted by the action in your answers. For example, if some X is the root of the given sample tree, you shouldn't always have isRoot(X) in your answers unless it is no longer the root after the action, and you **don't need** to have a negated one once the condition is no longer satisfied.
- Pay attention to the **order of the parameters** when there are multiple.

- In this question, a node can have only one parent but can have multiple children.

A. [5%] What are the current conditions for the given sample tree? Check all valid conditions below.

- ☐ isRoot(R)
- ☐ isRoot(B)
- ☐ isRoot(C)
- ☐ isRoot(D)
- ☐ isEmpty()
- ☐ pointTo(R, A)
- ☐ pointTo(R, B)
- ☐ pointTo(R, C)
- ☐ pointTo(R, D)
- ☐ pointTo(A, C)
- ☐ pointTo(A, D)
- ☐ pointTo(A, R)
- ☐ pointTo(B, R)
- ☐ pointTo(C, A)
- ☐ pointTo(C, R)
- ☐ pointTo(D, A)
- ☐ pointTo(D, R)

B. [4%] Please judge whether the following statements are true or false.

The pre and post conditions for action addChild(X, Y) are always the same under all situations (Assuming the tree is not empty before this action). T

The pre and post conditions for action removeNode(X) are always the same under all situations. F

C. [2%] In the given situation, what are the postconditions for action addChild(A, E). Check all valid options below (follow the requirements above).

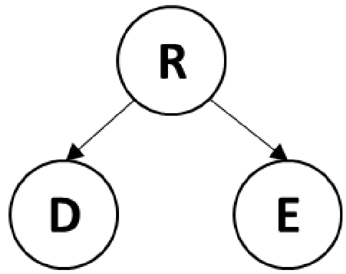
- ☐ isRoot(R)
- ☐ isRoot(B)
- ☐ isRoot(C)
- ☐ isRoot(D)
- ☐ isEmpty()
- ☐ pointTo(R, A)

- ☐ pointTo(R, B)
- ☐ pointTo(R, C)
- ☐ pointTo(R, D)
- ☐ pointTo(A, C)
- ☐ pointTo(A, D)
- ☐ pointTo(A, E)
- ☐ pointTo(A, R)
- ☐ pointTo(B, R)
- ☐ pointTo(C, A)
- ☐ pointTo(C, R)
- ☐ pointTo(C, E)
- ☐ pointTo(D, A)
- ☐ pointTo(D, E)
- ☐ pointTo(D, R)

D. [5%] In the given situation (before the action in question C), what are the preconditions for action removeNode(R). Check all valid options below (follow the requirements above).

- ☐ isRoot(R)
- ☐ isRoot(B)
- ☐ isRoot(C)
- ☐ isRoot(D)
- ☐ isEmpty()
- ☐ pointTo(R, A)
- ☐ pointTo(R, B)
- ☐ pointTo(R, C)
- ☐ pointTo(R, D)
- ☐ pointTo(A, C)
- ☐ pointTo(A, D)
- ☐ pointTo(A, R)
- ☐ pointTo(B, R)
- ☐ pointTo(C, A)
- ☐ pointTo(C, R)
- ☐ pointTo(D, A)
- ☐ pointTo(D, R)

E. [2%] To reach the following state, how many steps, in minimum, should be taken from the initial state?



4

F. [2%] Please judge whether the following statements are true or false.

In question E, linearization is not needed to get a valid plan. F

7. Multiple choice [10%]

1. [2.5%] Consider the universe of discourse to be the set of all nodes of directed graphs and let the atomic binary predicate symbol e stand for the edge relation on nodes, i.e. e(x, y) stands for there is an edge from node x to node y in a directed graph. Further, let “=” stand for the usual identity relation on nodes.
Which of the following **can be true** for a directed graph:

1. ∀x [∃y [~(x = y) ^ e(x, y)]]
2. ∀x [∀y [e(x, y) => ~(x = y)]]
3. ∀x [∀y [~(x = y) => (e(x, y) => e(y, x))]]
4. ∀x [∀y [∀z [e(x, y) ^ e(y, z) => e(x, z)]]]
5. ∀x ∀y ~(x = y) => [e(x, y) v e(y, x)]

2. [2.5%] If “Everyone in the world loves a lover”(interpreted as anyone who is a lover is loved by everyone in the world) and “Romeo loves Juliet” are true, then:

1. I love you
2. You love yourself
3. Everyone loves everyone
4. If I love you, then you love me
5. Dude, No one loves anyone.

Solution:
Romeo loves Juliet means Romeo is a lover.
All the world loves romeo. If all the world loves romeo, everyone in the world is a lover.
Everyone is a lover means, all the world loves everyone.

3. [2.5%] Given:

- => and <=> are both right associative meaning, X=>Y=>Z should be considered as (X => (Y => Z))
- A set of operators O is said to be adequate for propositional logic, if for every formula in propositional logic, there is a logically equivalent formula using only the operators in O.

Which of the following are true :

1. False |= True
2. X => X => X => X => X … (inf) is a Tautology
3. X => Y => X is a Tautology
4. We can unify P(x, y, F(z)) and Q(a, b, F(Madonna)).
5. {⋁, ¬} is an adequate set of operators for Propositional Logic

4. [2.5%] Given

- A set of operators O is said to be adequate for propositional logic, if for every formula in propositional logic, there is a logically equivalent formula using only the operators in O.
- Let $\Gamma = \{ \phi_i \mid 1 <= i <= n \}$ be a finite set of propositions, and let ϒ be any proposition.

Which of the following are true

1. { => , ¬} is an adequate set of operators for Propositional Logic
2. $\Gamma \models \Upsilon$ if and only if, $((\ldots((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \ldots \wedge \phi_n)) \Rightarrow \Upsilon$ is a tautology.
3. $\Gamma \models \Upsilon$ if and only if, $((\ldots((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \ldots \wedge \phi_n)) \wedge \neg\Upsilon$ is a contradiction.
4. $\Gamma \models \Upsilon$ if and only if, $((\ldots((\phi_1 \wedge \phi_2) \wedge \phi_3) \wedge \ldots \wedge \phi_n)) \vee \neg\Upsilon$ is a contradiction.
5. ((X => Y) => X) => X is a Tautology.