Unit 9 - Week 7

Assignment 7

The due date for submitting this assignment has passed.
As per our records you have not submitted this assignment.

Due on 2020-03-18, 23:59 IST.

FOL notation used in this assignment:

Predicates and constants are in CamelCase (first-caps, see Wikipedia); variables and functions are in camelCase (starts with lowercase).

"A implies B" takes one of four forms: $A \supset B$; $A \longrightarrow B$; $B \subset A$; $B \longleftarrow A$.

For example:

$\forall x\ \forall y\ [\ Edge(x, y) \supset Path(x, y)\ ]$

$\forall x\ \forall y\ [\ Edge(x, y) \longrightarrow Path(x, y)\ ]$

$\forall x\ \forall y\ [\ Path(x, y) \subset Edge(x, y)\ ]$

$\forall x\ \forall y\ [\ Path(x, y) \longleftarrow Edge(x, y)\ ]$

1) There are two baskets. Apples are kept in one basket and oranges in another basket. Bob states that “Ana eats ONLY apples”. In which of the following scenarios Bob’s statement is true according to FOL? *1 point*

- Ana ate all the apples and none of the oranges
- Ana ate all the apples and some oranges
- There are apples but Ana did not eat anything
- There are no apples and Ana did not eat anything
- There are no apples and Ana ate oranges
- Ana ate some apples and none of the oranges
- Ana ate some apples and some oranges

No, the answer is incorrect.
Score: 0
Accepted Answers:
Ana ate all the apples and none of the oranges
There are apples but Ana did not eat anything
There are no apples and Ana did not eat anything
Ana ate some apples and none of the oranges

2) There are two baskets. Apples are kept in one basket and oranges in another basket. Bob states that “Ana eats ALL the apples”. In which of the following scenarios Bob’s statement is true according to FOL? *1 point*

- Ana ate all the apples and none of the oranges
- Ana ate all the apples and some oranges
- There are apples but Ana did not eat anything
- There are no apples and Ana did not eat anything
- There are no apples and Ana ate oranges
- Ana ate some apples and none of the oranges
- Ana ate some apples and some oranges

No, the answer is incorrect.
Score: 0
Accepted Answers:
Ana ate all the apples and none of the oranges
Ana ate all the apples and some oranges
There are no apples and Ana did not eat anything
There are no apples and Ana ate oranges

3) Which of the following statements are TRUE about Forward Chaining: *1 point*

- It is sound for doing inference in propositional logic
- It is complete for doing inference in propositional logic
- It is sound for doing inference in first order logic
- It is complete for doing inference in first order logic
- It is sound for doing inference in first order definite clauses
- It is complete for doing inference in first order definite clauses

No, the answer is incorrect.
Score: 0
Accepted Answers:
It is sound for doing inference in propositional logic
It is complete for doing inference in propositional logic
It is sound for doing inference in first order logic
It is sound for doing inference in first order definite clauses
It is complete for doing inference in first order definite clauses

4) Which of the following statements are TRUE about Backward Chaining: *1 point*

- It is a goal-driven inference method
- It is complete for doing inference in propositional logic
- It is complete for doing inference in first order logic
- It is incomplete due to possibility of infinite loops
- Solving repeated subgoals make it inefficient

No, the answer is incorrect.
Score: 0
Accepted Answers:
It is a goal-driven inference method
It is incomplete due to possibility of infinite loops
Solving repeated subgoals make it inefficient

5) Which of the following statements are TRUE about the Resolution Method: *1 point*

- It is sound for propositional logic
- It is complete for propositional logic
- It is sound for first order logic
- It is complete for first order logic
- It is refutation-complete for first order logic

No, the answer is incorrect.
Score: 0
Accepted Answers:
It is sound for propositional logic
It is complete for propositional logic
It is sound for first order logic
It is refutation-complete for first order logic

6) The Resolution Refutation Method works by *1 point*

- refuting the resolution rule
- adding the negated goal to the set of clauses and showing that the set becomes unsatisfiable
- applying the strategy of proof by contradiction
- repeatedly applying the resolution rule on the sentences in the KB

No, the answer is incorrect.
Score: 0
Accepted Answers:
adding the negated goal to the set of clauses and showing that the set becomes unsatisfiable
applying the strategy of proof by contradiction
repeatedly applying the resolution rule on the sentences in the KB

7) Resolution-refutation for FOL is *1 point*

- Semi-decidable because if the set of clauses is unsatisfiable the method terminates with the null clause
- Semi-decidable because it may or may not terminate if the set of clause is not unsatisfiable
- Semi-decidable because it may or may not terminate irrespective of the satisfiability of conclusion
- Complete
- Sound
- Decidable because it derives null clause whenever the conclusion is unsatisfiable

No, the answer is incorrect.
Score: 0
Accepted Answers:
Semi-decidable because if the set of clauses is unsatisfiable the method terminates with the null clause
Semi-decidable because it may or may not terminate if the set of clause is not unsatisfiable
Complete
Sound

8) What is the clause form for $\neg\exists n\ \exists m\ [\ Red(n) \wedge Green(m) \wedge Path(n, m)\ ]$ *1 point*

- $\exists n\ \exists m\ [\ Red(n) \wedge Green(m) \wedge Path(n, m)\ ]$
- $\forall n\ \forall m\ \neg[\ Red(n) \wedge Green(m) \wedge Path(n, m)\ ]$
- $\forall n\ \forall m\ [\ \neg Red(n) \vee \neg Green(m) \vee \neg Path(n, m)\ ]$
- Clause form does not exist

No, the answer is incorrect.
Score: 0
Accepted Answers:
$\forall n\ \forall m\ [\ \neg Red(n) \vee \neg Green(m) \vee \neg Path(n, m)\ ]$

9) Consider the following FOL sentences: *1 point*

A. $\forall x\ \forall y\ [\ Path(x, y) \leftarrow Edge(x, y)\ ]$

B. $\forall u\ \forall w\ [\ Path(u, w) \leftarrow \exists v\ (\ Edge(u, v) \wedge Path(v, w)\ )\ ]$

Given no other information, what can you say about Edge and Path relations?

- Edge is a binary relation
- Path is a binary relation
- Edge is a symmetric relation
- Edge is a transitive relation
- Path is a symmetric relation
- Path is a transitive relation
- Edge relation is a subset of Path relation

No, the answer is incorrect.
Score: 0
Accepted Answers:
Edge is a binary relation
Path is a binary relation
Path is a transitive relation
Edge relation is a subset of Path relation

Convert the following FOL sentences into clause form.

A. $\forall x\ \forall y\ [\ Path(x, y) \leftarrow Edge(x, y)\ ]$
B. $\forall u\ \forall w\ [\ Path(u, w) \leftarrow \exists v\ (\ Edge(u, v) \wedge Path(v, w)\ )\ ]$
C. $\neg\exists n\ \exists m\ [\ Red(n) \wedge Green(m) \wedge Path(n, m)\ ]$
D. $\neg\exists n\ \exists m\ [\ (\ Red(n) \vee Green(m)\ ) \wedge Path(n, m)\ ]$

Match the clause form given below to the the FOL formula above. Enter the item label (one of A, B, C, D) in the textbox. If there is no matching formula type NONE.

10) { ¬Edge(?x, ?y), Path(?x, ?y) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) A

1 point

11) { ¬Edge(?u, ?v), ¬Path(?v, ?w), Path(?u, ?w) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) B

1 point

12) { ¬Edge(?u, skV(?u, ?w)), ¬Path(skV(?u, ?w), ?w), Path(?u, ?w) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) NONE

1 point

13) { ¬Red(?n), ¬Green(?m), ¬Path(?n, ?m) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) C

1 point

14) { Red(?n), Green(?m), Path(?n, ?m) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) NONE

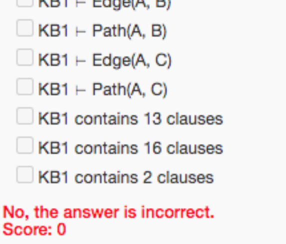
1 point

15) { ¬Red(?n), ¬Path(?n, ?m) }; { ¬Green(?m), ¬Path(?n, ?m) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
(Type: String) D

1 point

16) A knowledge base KB1 that describes a directed network is given in clause form. It contains six constants { A, B, C, D, E, F }, three unary predicates { Red, Green, Blue } and two binary predicates { Edge, Path }. Edge(x,y) represents a directed edge from node x to node y. Path(x,y) indicates the existence of a route (one or more directed edges) from node x to node y. Path does not give us the route but indicates its existence. (Hint: Encode KB1 in Prolog and experiment with it.) *1 point*



KB1 = {
{ ¬Edge(?x, ?y), Path(?x, ?y) };
{ ¬Edge(?u, ?v), ¬Path(?v, ?w), Path(?u, ?w) };
Red(A); Blue(B); Blue(C); Green(D); Blue(E); Blue(F);
Edge(A,C); Edge(B,C); Edge(C,D); Edge(D,E); Edge(D,F);
}

Which of the following are true?

- The first two clauses in KB1 are derived from the two FOL sentences in Question 9
- KB1 ⊢ Edge(A, B)
- KB1 ⊢ Path(A, B)
- KB1 ⊢ Edge(A, C)
- KB1 ⊢ Path(A, C)
- KB1 contains 13 clauses
- KB1 contains 16 clauses
- KB1 contains 2 clauses

No, the answer is incorrect.
Score: 0
Accepted Answers:
The first two clauses in KB1 are derived from the two FOL sentences in Question 9
KB1 ⊢ Edge(A, C)
KB1 ⊢ Path(A, C)
KB1 contains 13 clauses

17) Construct KB2 from KB1 in previous question. Take KB1, resolve each Edge in KB1 with the the first clause { ¬Edge(?x, ?y), Path(?x, ?y) } in KB1. It will result in new facts. *1 point*
Collect the clauses from KB1 and the newly generated facts and store in KB2. Now, KB2 is equal to:

- KB1 union { Edge(A, C); Edge(B, C); Edge(C, D); Edge(D, E); Edge(D, F) }
- KB1 union { Edge(A, D); Edge(B, D); Edge(C, E); Edge(C, F) }
- KB1 union { Edge(A, E); Edge(A, F); Edge(B, E); Edge(B, F) }
- KB1 union { Path(A, C); Path(B, C); Path(C, D); Path(D, E); Path(D, F) }
- KB1 union { Path(A, D); Path(B, D); Path(C, E); Path(C, F) }
- KB1 union { Path(A, E); Path(A, F); Path(B, E); Path(B, F) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
KB1 union { Path(A, C); Path(B, C); Path(C, D); Path(D, E); Path(D, F) }

18) Construct KB3 from KB2 in previous question. Take KB2, resolve the second clause { ¬Edge(?u, ?v), ¬Path(?v, ?w), Path(?u, ?w) } in KB2 with the Edges and Paths in KB2. It will result in new facts. *1 point*
Collect the clauses from KB2 and the newly generated facts and store in KB3. Now, KB3 is equal to:

- KB2 union { Edge(A, C); Edge(B, C); Edge(C, D); Edge(D, E); Edge(D, F) }
- KB2 union { Edge(A, D); Edge(B, D); Edge(C, E); Edge(C, F) }
- KB2 union { Edge(A, E); Edge(A, F); Edge(B, E); Edge(B, F) }
- KB2 union { Path(A, C); Path(B, C); Path(C, D); Path(D, E); Path(D, F) }
- KB2 union { Path(A, D); Path(B, D); Path(C, E); Path(C, F) }
- KB2 union { Path(A, E); Path(A, F); Path(B, E); Path(B, F) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
KB2 union { Path(A, D); Path(B, D); Path(C, E); Path(C, F) }

19) Construct KB4 from KB3 in previous question. Take KB3, resolve the second clause { ¬Edge(?u, ?v),¬Path(?v, ?w), Path(?u, ?w) } in KB3 with the Edges and NEWLY ADDED Paths in KB3. It will result in new facts. (Hint: step by step generation of paths can be emulated in Prolog, check the forum.) *1 point*

Collect the clauses from KB3 and the newly generated facts and store in KB4. Now, KB4 is equal to:

- KB3 union { Edge(A, C); Edge(B, C); Edge(C, D); Edge(D, E); Edge(D, F) }
- KB3 union { Edge(A, D); Edge(B, D); Edge(C, E); Edge(C, F) }
- KB3 union { Edge(A, E); Edge(A, F); Edge(B, E); Edge(B, F) }
- KB3 union { Path(A, C); Path(B, C); Path(C, D); Path(D, E); Path(D, F) }
- KB3 union { Path(A, D); Path(B, D); Path(C, E); Path(C, F) }
- KB3 union { Path(A, E); Path(A, F); Path(B, E); Path(B, F) }

No, the answer is incorrect.
Score: 0
Accepted Answers:
KB3 union { Path(A, E); Path(A, F); Path(B, E); Path(B, F) }

20) Given KB1, KB4 was derived by repeatedly resolving clauses. What is true about KB4? *1 point*

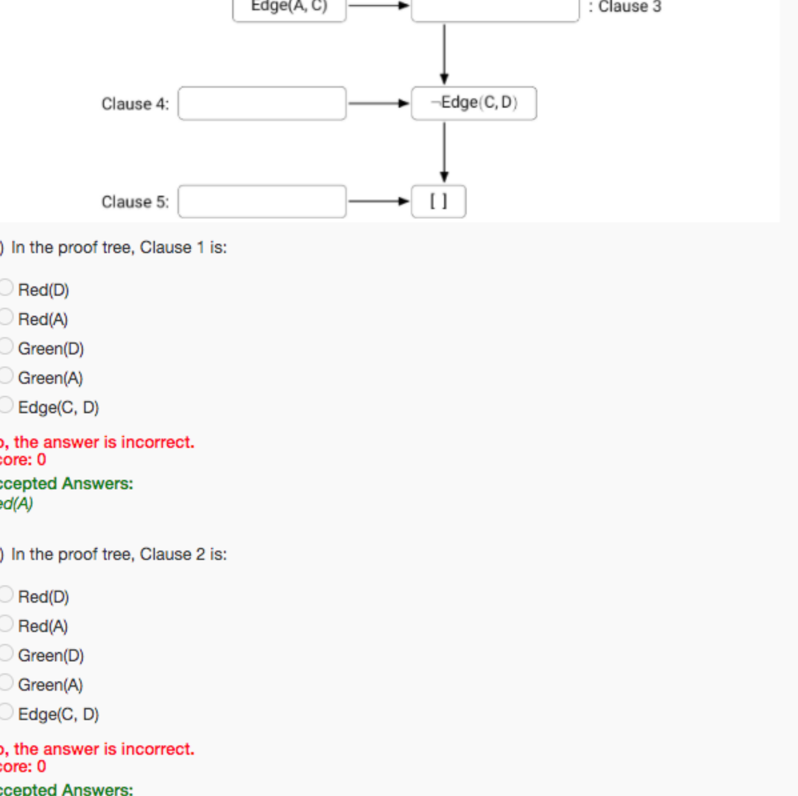
- KB4 $\models$ [Red(A) $\wedge$ Path(A, D) $\wedge$ Green(D)]
- KB4 has 5 Edge facts
- KB4 has 13 Path facts
- KB4 cannot be expanded further, it contains all the facts
- In KB4, queries can be answered by a direct lookup on facts (Red, Green, Blue, Edge and Path).

No, the answer is incorrect.
Score: 0
Accepted Answers:
KB4 $\models$ [Red(A) $\wedge$ Path(A, D) $\wedge$ Green(D)]
KB4 has 5 Edge facts
KB4 has 13 Path facts
KB4 cannot be expanded further, it contains all the facts
In KB4, queries can be answered by a direct lookup on facts (Red, Green, Blue, Edge and Path).

BEGIN GROUP (2)

A proof by Resolution Refutation is shown for the query "$\exists n\ \exists m\ [\ Red(n) \wedge Path(n, m) \wedge Green(m)\ ]$" over the knowledge base KB1 from Question 16.

Find the missing elements in the proof tree.



21) In the proof tree, Clause 1 is: *1 point*

- Red(D)
- Red(A)
- Green(D)
- Green(A)
- Edge(C, D)

No, the answer is incorrect.
Score: 0
Accepted Answers:
Red(A)

22) In the proof tree, Clause 2 is: *1 point*

- Red(D)
- Red(A)
- Green(D)
- Green(A)
- Edge(C, D)

No, the answer is incorrect.
Score: 0
Accepted Answers:
Green(D)

23) In the proof tree, Clause 3 is: *1 point*

- { ¬Edge(?x, ?y), Path(?x, ?y) }
- { ¬Edge(?u, ?v), ¬Path(?v, ?w), Path(?u, ?w) }
- Edge(C, D)
- ¬Path(C, D)
- Path(C, D)

No, the answer is incorrect.
Score: 0
Accepted Answers:
¬Path(C, D)

24) In the proof tree, Clause 4 is: *1 point*

- {¬Edge(?x, ?y), Path(?x, ?y) }
- { ¬Edge(?u, ?v), ¬Path(?v, ?w), Path(?u, ?w) }
- Edge(C, D)
- ¬Path(C, D)
- Path(C, D)

No, the answer is incorrect.
Score: 0
Accepted Answers:
{¬Edge(?x, ?y), Path(?x, ?y) }

25) In the proof tree, Clause 5 is: *1 point*

- { ¬Edge(?x, ?y), Path(?x, ?y) }
- Edge(C, D)
- Edge(D, C)
- ¬Path(C, D)
- Path(C, D)

No, the answer is incorrect.
Score: 0
Accepted Answers:
Edge(C, D)

26) Identify the bindings for the variables in the proof tree. *1 point*

- ?n=D; ?m=A; ?u=A; ?w=D; ?v=D;
- ?n=A; ?m=D; ?u=A; ?w=D; ?v=C;
- ?n=A; ?m=A; ?u=A; ?w=D; ?v=B;
- ?n=D; ?m=D; ?u=A; ?w=D; ?v=C;

No, the answer is incorrect.
Score: 0
Accepted Answers:
?n=A; ?m=D; ?u=A; ?w=D; ?v=C;

END GROUP (2)