**Chapter 9**

**Predicate Logic**

The chapter consist of Short type Questions &Answers , Descriptive Question & Answer and MCQs & answers

Contents

[Short type Questions &Answers 3](#_Toc14943735)

[Q Discuss First order logic with example? 3](#_Toc14943736)

[Q Define semantic network? 3](#_Toc14943737)

[Q Define Horn clause? 3](#_Toc14943738)

[Q Analyse clausal form and its usefulness 3](#_Toc14943739)

[Q Explain some knowledge representation techniques? 4](#_Toc14943740)

[Q Differentiate propositional and First order logic. 4](#_Toc14943741)

[Q Compose the meaning of resolution/Refutation 4](#_Toc14943742)

[Q How predicate logic is helpful in knowledge representation? 5](#_Toc14943743)

[Q Compose a well formed formula? 5](#_Toc14943744)

[Descriptive Question & Answer 5](#_Toc14943745)

[Q Consider the sentence "Hammers are for driving nails into surfaces." Name two words in this sentence that are lexically ambiguous. (There are at least four.) For each of these two words, describe a disambiguation technique which will choose the right interpretation over at least one of the wrong interpretations. Be specific. 5](#_Toc14943746)

[Here, I think, one has to rely on world knowledge that a hammer is used to drive a nail through the surface of an object into its interior. 6](#_Toc14943747)

[Q Explain Syntax and semantics for Knowledge Representation 6](#_Toc14943748)

[Q What is resolution Explain 7](#_Toc14943749)

[Resolution 7](#_Toc14943750)

[Q Consider the following axioms: 8](#_Toc14943751)

[All hounds howl at night. 8](#_Toc14943752)

[Anyone who has any cats will not have any mice. 8](#_Toc14943753)

[Light sleepers do not have anything which howls at night. 8](#_Toc14943754)

[John has either a cat or a hound. 8](#_Toc14943755)

[(Conclusion) If John is a light sleeper, then John does not have any mice. 8](#_Toc14943756)

[Prove the conclusion using resolution 8](#_Toc14943757)

[Q Explain resolution in predicate logic using example 10](#_Toc14943758)

[Q Explain the concept of Unification 13](#_Toc14943759)

[Q Consider the following axioms: 16](#_Toc14943760)

[Every child loves Santa. 16](#_Toc14943761)

[Everyone who loves Santa loves any reindeer. 16](#_Toc14943762)

[Rudolph is a reindeer, and Rudolph has a red nose. 16](#_Toc14943763)

[Anything which has a red nose is weird or is a clown. 16](#_Toc14943764)

[No reindeer is a clown. 16](#_Toc14943765)

[Scrooge does not love anything which is weird. 16](#_Toc14943766)

[(Conclusion) Scrooge is not a child. 16](#_Toc14943767)

[Represent these axioms in predicate calculus; skolemize as necessary and convert each formula to clause form. (Note: `has a red nose' can be a single predicate. Remember to negate the conclusion.) Prove the unsatisfiability of the set of clauses by resolution. 16](#_Toc14943768)

[Consider the following axioms: 17](#_Toc14943769)

[Anyone who buys carrots by the bushel owns either a rabbit or a grocery store. 17](#_Toc14943770)

[Every dog chases some rabbit. 17](#_Toc14943771)

[Mary buys carrots by the bushel. 17](#_Toc14943772)

[Anyone who owns a rabbit hates anything that chases any rabbit. 17](#_Toc14943773)

[John owns a dog. 17](#_Toc14943774)

[Someone who hates something owned by another person will not date that person. 17](#_Toc14943775)

[(Conclusion) If Mary does not own a grocery store, she will not date John. 17](#_Toc14943776)

[Represent these clauses in predicate calculus, using only those predicates which are necessary. For example, you need not represent `person', and phrases such as `who buys carrots by the bushel' may be represented by a single predicate. Negate the conclusion and convert to clause form, skolemizing as necessary. Prove the unsatisfiability of the resulting set of clauses by resolution. 17](#_Toc14943777)

[Consider the following axioms: 18](#_Toc14943778)

[Every Austinite who is not conservative loves some armadillo. 18](#_Toc14943779)

[Anyone who wears maroon-and-white shirts is an Aggie. 18](#_Toc14943780)

[Every Aggie loves every dog. 18](#_Toc14943781)

[Nobody who loves every dog loves any armadillo. 18](#_Toc14943782)

[Clem is an Austinite, and Clem wears maroon-and-white shirts. 18](#_Toc14943783)

[(Conclusion) Is there a conservative Austinite? 18](#_Toc14943784)

[. Consider the following axioms: 19](#_Toc14943785)

[Every coyote chases some roadrunner. 19](#_Toc14943786)

[Every roadrunner who says ``beep-beep'' is smart. 19](#_Toc14943787)

[No coyote catches any smart roadrunner. 19](#_Toc14943788)

[Any coyote who chases some roadrunner but does not catch it is frustrated. 19](#_Toc14943789)

[(Conclusion) If all roadrunners say ``beep-beep'', then all coyotes are frustrated. 19](#_Toc14943790)

[MCQs & answers 19](#_Toc14943791)

# Short type Questions &Answers

## Q Discuss First order logic with example?

First-order logic is symbolized reasoning in which each sentence, or statement, is broken down into a subject and a predicate. The predicate modifies or defines the properties of the subject. In first-order logic, a predicate can only refer to a single subject. First-order logic is also known as first-order predicate calculus or first-order functional calculus. In first-order logic, a sentence can be structured using the universal quantifier (symbolized ) or the existential quantifier

## Q Define semantic network?

A semantic network, or frame network, is a network that represents semanticrelations between concepts. This is often used as a form of knowledge representation. It is a directed or undirected graph consisting of vertices, which represent concepts, and edges.

## Q Define Horn clause?

A *Horn clause*is a clause with at most one positive literal.   
Any Horn clause therefore belongs to one of four categories:

A *rule*: 1 positive literal, at least 1 negative literal

A *fact*or *unit*: 1 positive literal, 0 negative literals.

A *negated goal*: 0 positive literals, at least 1 negative literal.

The null clause: 0 positive and 0 negative literals. Appears only as the end of a resolution proof.

## Q Analyse clausal form and its usefulness

To demonstrate that assumptions imply a conclusion, it is helpful to construct a proof consisting of inference steps. For the proof to be convincing, the steps need to be direct and obvious, so the sentences should be unambiguous and their grammar should be as simple as possible.

The clausal form of logic fits this simplicity requirement. Its simplest sentences are*atomic sentences*, which name relationships between individuals:

likes(sweetie, rover)  
it\_is\_raining.  
gives(eve, adam, apple).

## Q Explain some knowledge representation techniques?

●     There are different ways of representing knowledge e.g.

predicate logic,

semantic networks,

extended semantic net,

frames,

      conceptual dependency

## Q Differentiate propositional and First order logic.

Propositional logic is declarative.

It allows partial/disjunctive/negative information.

Propositional logic has very limited expressive power.

FOL

It deals with object,relation and function.

It uses connectives

The FOL has more expressive power.

## Q Compose the meaning of resolution/Refutation

Resolution yields a complete inference algorithm when coupled with any complete search algorithm.Resolution makes use of the inference rules.Resolution performs deductive inference.Resolution uses proof by contradiction. One can perform Resolution from a Knowledge Base. A Knowledge Base is a collection of facts or one can even

call it a database with all facts.

## Q How predicate logic is helpful in knowledge representation?

predicate logic, involves using standard forms of logical symbolism which have been familiar to philosophers and mathematicians for many decades. Most simple sentences, for example, ``Peter is generous'' or ``Jane gives a painting to Sam,'' can be represented in terms of logical formulae in which a predicate is applied to one or morearguments

## Q Compose a well formed formula?

Not all strings can represent propositions of the predicate logic. Those which produce a proposition when their symbols are interpreted must follow the rules given below, and they are called [wffs](https://www.blogger.com/null)(well-formed formulas) of the first order predicate logic.

# Descriptive Question & Answer

## Q Consider the sentence "Hammers are for driving nails into surfaces." Name two words in this sentence that are lexically ambiguous. (There are at least four.) For each of these two words, describe a disambiguation technique which will choose the right interpretation over at least one of the wrong interpretations. Be specific.

Answer:

* A. The preposition "for" has many different meanings. Even in the phrase "are for" it can mean:
  + i. "are used for" as above.
  + ii. "favors" as in "Cheney is for burning more fossil fuel and against conserving resources." This can be ruled out by selectional restrictions: this meaning requires an animate subject, unlike "hammers".
  + iii. "are intended to be given to" as in "The presents are for the baby." This can be ruled out by selectional restrictions: it requires an animate object, unlike "driving".
* B. "driving" can mean:
  + i. forcing an object to move against resistance, as above.
  + ii. driving a car (by far the most frequent meaning).
  + iii. impelling a person to undesired behaviors (as in "driving me crazy", "driving me to drink")
  + quite a few other specialized meanings (iv. "driving" in golf, v. "driving cattle", etc.)

However, most of these can be ruled out by selectional restrictions on the object. E.g. (ii) requires a car as object; (iii) and (v) require animate objects, etc.

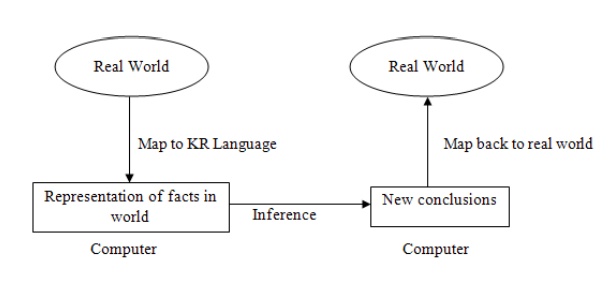
* C. "nails" can be either the tool or fingernails. However, frequency in the context of "hammer" gives a preference for the tool.
* D. "into" can mean
  + i. motion into the interior of a region, as in "He drove the nail into the board".
  + ii. motion through a boundary into the interior of another region, as above.
  + iii. "in the direction of" as in "He looked into the sun".
  + iv. "against" as in "He ran into a brick wall".

## Here, I think, one has to rely on world knowledge that a hammer is used to drive a nail through the surface of an object into its interior.

## Q Explain Syntax and semantics for Knowledge Representation

Knowledge representation languages should have precise syntax and semantics. You must know exactly what an expression means in terms of objects in the real world. Suppose we have decided that “red 1” refers to a dark red colour, “car1” is my car, car2 is another. Syntax of language will tell you which of the following is legal: red1 (car1), red1 car1, car1 (red1), red1 (car1 & car2)?

Semantics of language tell you exactly what an expression means: for example, Pred (Arg) means that the property referred to by Pred applies to the object referred to by Arg. E.g., properly “dark red” appli es to my car.



## Q What is resolution Explain

### Resolution

Resolution yields a complete inference algorithm when coupled with any complete search algorithm.Resolution makes use of the inference rules. Resolution performs deductive inference.Resolution uses proof by contradiction. One can perform Resolution from a Knowledge Base. A Knowledge Base is a collection of facts or one can even call it a database with all facts.

Resolution Algorithm:

Resolution basically works by using the principle of proof by contradiction. To find the conclusion we should negate the conclusion. Then the resolution rule is applied to the resulting clauses. Each clause that contains complementary literals is resolved to produce a two new clause,which can be added to the set of facts (if it is not already present).This process continues until one of the two things happen:

·         There are no new clauses that can be added

·         An application of the resolution rule derives the empty clause

An empty clause shows that the negation of the conclusion is a complete contradiction ,hence the negation of the conclusion is invalid or false or the assertion is completely valid or true.

Steps for Resolution Refutation

    1. Convert all the propositions of KB to clause form (S).  
2. Negate a and convert it to clause form. Add it to S.  
3. Repeat until either a contradiction is found or no progress can  
be made:

## Q Consider the following axioms:

## All hounds howl at night.

## Anyone who has any cats will not have any mice.

## Light sleepers do not have anything which howls at night.

## John has either a cat or a hound.

## (Conclusion) If John is a light sleeper, then John does not have any mice.

## Prove the conclusion using resolution

The conclusion can be proved using Resolution as shown below. The first step is to write each axiom as a well-formed formula in first-order predicate calculus. The clauses written for the above axioms are shown below, using LS(x) for `light sleeper'.

1. *∀ x (HOUND(x) → HOWL(x))*
2. *∀ x ∀ y (HAVE (x,y) ∧ CAT (y) → ¬ ∃ z (HAVE(x,z) ∧ MOUSE (z)))*
3. *∀ x (LS(x) → ¬ ∃ y (HAVE (x,y) ∧ HOWL(y)))*
4. *∃ x (HAVE (John,x) ∧ (CAT(x) ∨ HOUND(x)))*
5. *LS(John) → ¬ ∃ z (HAVE(John,z) ∧ MOUSE(z))*

The next step is to transform each wff into Prenex Normal Form, skolemize, and rewrite as clauses in conjunctive normal form; these transformations are shown below.

1. *∀ x (HOUND(x) → HOWL(x))*

*¬ HOUND(x) ∨ HOWL(x)*

1. *∀ x ∀ y (HAVE (x,y) ∧ CAT (y) &rarr ¬ ∃ z (HAVE(x,z) ∧ MOUSE (z)))*

*∀ x ∀ y (HAVE (x,y) ∧ CAT (y) &rarr ∀ z ¬ (HAVE(x,z) ∧ MOUSE (z)))*

*∀ x ∀ y ∀ z (¬ (HAVE (x,y) ∧ CAT (y)) ∨ ¬ (HAVE(x,z) ∧ MOUSE (z)))*

*¬ HAVE(x,y) ∨ ¬ CAT(y) ∨ ¬ HAVE(x,z) ∨ ¬ MOUSE(z)*

1. *∀ x (LS(x) &rarr ¬ ∃ y (HAVE (x,y) ∧ HOWL(y)))*

*∀ x (LS(x) → ∀ y ¬ (HAVE (x,y) ∧ HOWL(y)))*

*∀ x ∀ y (LS(x) → ¬ HAVE(x,y) ∨ ¬ HOWL(y))*

*∀ x ∀ y (¬ LS(x) ∨ ¬ HAVE(x,y) ∨ ¬ HOWL(y))*

*¬ LS(x) ∨ ¬ HAVE(x,y) ∨ ¬ HOWL(y)*

1. *∃ x (HAVE (John,x) ∧ (CAT(x) ∨ HOUND(x)))*

*HAVE(John,a) ∧ (CAT(a) ∨ HOUND(a))*

1. *¬ [LS(John) → ¬ ∃ z (HAVE(John,z) ∧ MOUSE(z))] (negated conclusion)*

*¬ [¬ LS (John) ∨ ¬ ∃ z (HAVE (John, z) ∧ MOUSE(z))]*

*LS(John) ∧ ∃ z (HAVE(John, z) ∧ MOUSE(z)))*

*LS(John) ∧ HAVE(John,b) ∧ MOUSE(b)*

The set of CNF clauses for this problem is thus as follows:

1. *¬ HOUND(x) ∨ HOWL(x)*
2. *¬ HAVE(x,y) ∨ ¬ CAT(y) ∨ ¬ HAVE(x,z) ∨ ¬ MOUSE(z)*
3. *¬ LS(x) ∨ ¬ HAVE(x,y) ∨ ¬ HOWL(y)*
   1. *HAVE(John,a)*
   2. *CAT(a) ∨ HOUND(a)*
   3. *LS(John)*
   4. *HAVE(John,b)*
   5. *MOUSE(b)*

Now we proceed to prove the conclusion by resolution using the above clauses. Each result clause is numbered; the numbers of its parent clauses are shown to its left.

|  |  |  |
| --- | --- | --- |
| [1.,4.(b):] | 6. | *CAT(a) ∨ HOWL(a)* |
| [2,5.(c):] | 7. | *¬ HAVE(x,y) ∨ ¬ CAT(y) ∨ ¬ HAVE(x,b)* |
| [7,5.(b):] | 8. | *¬ HAVE(John,y) ∨ ¬ CAT(y)* |
| [6,8:] | 9. | *¬ HAVE(John,a) ∨ HOWL(a)* |
| [4.(a),9:] | 10. | *HOWL(a)* |
| [3,10:] | 11. | *¬ LS(x) ∨ ¬ HAVE(x,a)* |
| [4.(a),11:] | 12. | *¬ LS(John)* |
| [5.(a),12:] | 13. | *□* |

## Q Explain resolution in predicate logic using example

Two literals are contradictory if one can be unified with the negation of the other. For example man(x) and man (Himalayas) are contradictory since man(x) and man(Himalayas ) can be unified. In predicate logic unification algorithm is used to locate pairs of literals that cancel out. It is important that if two instances of the same variable occur, then they must be given identical substitutions. The resolution algorithm for predicate logic as follows

Let f be a set of given statements and S is a statement to be proved.

1. Covert all the statements of F to clause form.

2. Negate S and convert the result to clause form. Add it to the set of clauses obtained in 1.

3. Repeat until either a contradiction is found or no progress can be made or a predetermined amount of effort has been expended.

a) Select two clauses. Call them parent clauses.

b) Resolve them together. The resolvent will be the disjunction of all of these literals of both clauses. If there is a pair of literals T1 and T2 such that one parent clause contains Ti and the other contains T2 and if T1 and T2 are unifiable, then neither t1 nor T2 should appear in the resolvent. Here Ti and T2 are called complimentary literals.

If the resolvent is the empty clause , then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

Using resolution to produce proof is illustrated in the following statements.

1. Marcus was a man.

2. Marcus was a Pompeian

3. All pompeians were Romans

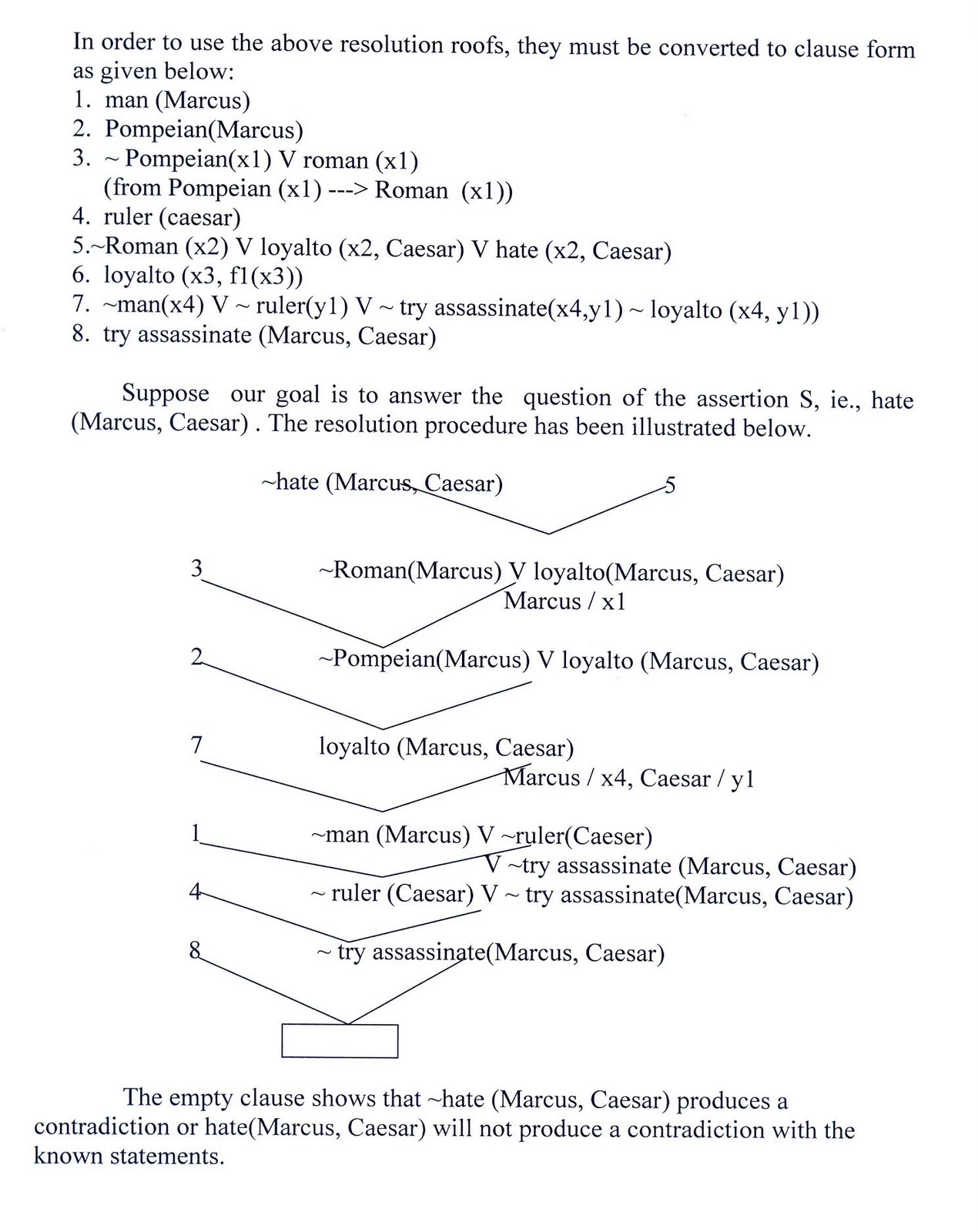
4. Caesar was a ruler

5. All Romans were either loyal to Caesar or hated him.

6. Everyone is loyal to someone.

7. People only try to assassinate rulers they are not loyal to.

8. Marcus tried to assassinate Caesar.



## Q Explain the concept of Unification

In propositional logic it is easy to determine that two literals can not both be true at the same time. Simply look for L and ~L . In predicate logic, this matching process is more complicated, since bindings of variables must be considered.

For example man (john) and man(john) is a contradiction while man (john) and man(Himalayas) is not. Thus in order to determine contradictions we need a matching procedure that compares two literals and discovers whether there exist a set of substitutions that makes them identical . There is a recursive procedure that does this matching . It is called Unification algorithm.

In Unification algorithm each literal is represented as a list, where first element is the name of a predicate and the remaining elements are arguments. The argument may be a single element (atom) or may be another list. For example we can have literals as

The Unification algorithm is listed below as a procedure UNIFY (L1, L2). It returns a list representing the composition of the substitutions that were performed during the match. An empty list NIL indicates that a match was found without any substitutions. If the list contains a single value F, it indicates that the unification procedure failed.

UNIFY (L1, L2)

1. if L1 or L2 is an atom part of same thing do

(a) if L1 or L2 are identical then return NIL

(b) else if L1 is a variable then do

(i) if L1 occurs in L2 then return F else return (L2/L1)

 else if L2 is a variable then do

(i) if L2 occurs in L1 then return F else return (L1/L2)

else return F.

2. If length (L!) is not equal to length (L2) then return F.

3. Set SUBST to NIL

( at the end of this procedure , SUBST will contain all the substitutions used to unify L1 and L2).

4. For I = 1 to number of elements in L1 do

i) call UNIFY with the i th element of L1 and I’th element of L2, putting the result in S

ii) if S = F then return F

iii) if S is not equal to NIL then do

(A) apply S to the remainder of both L1 and L2

(B) SUBST := APPEND (S, SUBST) return SUBST.

## Q Consider the following axioms:

## Every child loves Santa.

## Everyone who loves Santa loves any reindeer.

## Rudolph is a reindeer, and Rudolph has a red nose.

## Anything which has a red nose is weird or is a clown.

## No reindeer is a clown.

## Scrooge does not love anything which is weird.

## (Conclusion) Scrooge is not a child.

## Represent these axioms in predicate calculus; skolemize as necessary and convert each formula to clause form. (Note: `has a red nose' can be a single predicate. Remember to negate the conclusion.) Prove the unsatisfiability of the set of clauses by resolution.

 Consider the following axioms:

1. Every child loves Santa.   
   *∀ x (CHILD(x) → LOVES(x,Santa))*
2. Everyone who loves Santa loves any reindeer.   
   *∀ x (LOVES(x,Santa) → ∀ y (REINDEER(y) → LOVES(x,y)))*
3. Rudolph is a reindeer, and Rudolph has a red nose.   
   *REINDEER(Rudolph) ∧ REDNOSE(Rudolph)*
4. Anything which has a red nose is weird or is a clown.   
   *∀ x (REDNOSE(x) → WEIRD(x) ∨ CLOWN(x))*
5. No reindeer is a clown.   
   *¬ ∃ x (REINDEER(x) ∧ CLOWN(x))*
6. Scrooge does not love anything which is weird.   
   *∀ x (WEIRD(x) → ¬ LOVES(Scrooge,x))*
7. (Conclusion) Scrooge is not a child.   
   *¬ CHILD(Scrooge)*

## Consider the following axioms:

## Anyone who buys carrots by the bushel owns either a rabbit or a grocery store.

## Every dog chases some rabbit.

## Mary buys carrots by the bushel.

## Anyone who owns a rabbit hates anything that chases any rabbit.

## John owns a dog.

## Someone who hates something owned by another person will not date that person.

## (Conclusion) If Mary does not own a grocery store, she will not date John.

## Represent these clauses in predicate calculus, using only those predicates which are necessary. For example, you need not represent `person', and phrases such as `who buys carrots by the bushel' may be represented by a single predicate. Negate the conclusion and convert to clause form, skolemizing as necessary. Prove the unsatisfiability of the resulting set of clauses by resolution.

Consider the following axioms:

1. Anyone who buys carrots by the bushel owns either a rabbit or a grocery store.   
   *∀ x (BUY(x) → ∃ y (OWNS(x,y) ∧ (RABBIT(y) ∨ GROCERY(y))))*
2. Every dog chases some rabbit.   
   *∀ x (DOG(x) → ∃ y (RABBIT(y) ∧ CHASE(x,y)))*
3. Mary buys carrots by the bushel.   
   *BUY(Mary)*
4. Anyone who owns a rabbit hates anything that chases any rabbit.   
   *∀ x ∀ y (OWNS(x,y) ∧ RABBIT(y) → ∀ z ∀ w (RABBIT(w) ∧ CHASE(z,w) → HATES(x,z)))*
5. John owns a dog.   
   *∃ x (DOG(x) ∧ OWNS(John,x))*
6. Someone who hates something owned by another person will not date that person.   
   *∀ x ∀ y ∀ z (OWNS(y,z) ∧ HATES(x,z) → ¬ DATE(x,y))*
7. (Conclusion) If Mary does not own a grocery store, she will not date John.   
   *(( ¬ ∃ x (GROCERY(x) ∧ OWN(Mary,x))) → ¬ DATE(Mary,John))*

## Consider the following axioms:

## Every Austinite who is not conservative loves some armadillo.

## Anyone who wears maroon-and-white shirts is an Aggie.

## Every Aggie loves every dog.

## Nobody who loves every dog loves any armadillo.

## Clem is an Austinite, and Clem wears maroon-and-white shirts.

## (Conclusion) Is there a conservative Austinite?

1. Every Austinite who is not conservative loves some armadillo.   
   *∀ x (AUSTINITE(x) ∧ ¬ CONSERVATIVE(x) → ∃ y (ARMADILLO(y) ∧ LOVES(x,y)))*
2. Anyone who wears maroon-and-white shirts is an Aggie.   
   *∀ x (WEARS(x) → AGGIE(x))*
3. Every Aggie loves every dog.   
   *∀ x (AGGIE(x) → ∀ y (DOG(y) → LOVES(x,y)))*
4. Nobody who loves every dog loves any armadillo.   
   *¬ ∃ x ((∀ y (DOG(y) → LOVES(x,y))) ∧ ∃ z (ARMADILLO(z) ∧ LOVES(x,z)))*
5. Clem is an Austinite, and Clem wears maroon-and-white shirts.   
   *AUSTINITE(Clem) ∧ WEARS(Clem)*
6. (Conclusion) Is there a conservative Austinite?   
   *∃ x (AUSTINITE(x) ∧ CONSERVATIVE(x))*

( ( (not (Austinite x)) (Conservative x) (Armadillo (f x)) )

( (not (Austinite x)) (Conservative x) (Loves x (f x)) )

( (not (Wears x)) (Aggie x) )

( (not (Aggie x)) (not (Dog y)) (Loves x y) )

( (Dog (g x)) (not (Armadillo z)) (not (Loves x z)) )

( (not (Loves x (g x))) (not (Armadillo z)) (not (Loves x z)) )

( (Austinite (Clem)) )

( (Wears (Clem)) )

( (not (Conservative x)) (not (Austinite x)) ) )

## . Consider the following axioms:

## Every coyote chases some roadrunner.

## Every roadrunner who says ``beep-beep'' is smart.

## No coyote catches any smart roadrunner.

## Any coyote who chases some roadrunner but does not catch it is frustrated.

## (Conclusion) If all roadrunners say ``beep-beep'', then all coyotes are frustrated.

1. Every coyote chases some roadrunner.   
   *∀ x (COYOTE(x) → ∃ y (RR(y) ∧ CHASE(x,y)))*
2. Every roadrunner who says ``beep-beep'' is smart.   
   *∀ x (RR(x) ∧ BEEP(x) → SMART(x))*
3. No coyote catches any smart roadrunner.   
   *¬ ∃ x ∃ y (COYOTE(x) ∧ RR(y) ∧ SMART(y) ∧ CATCH(x,y))*
4. Any coyote who chases some roadrunner but does not catch it is frustrated.   
   *∀ x (COYOTE(x) ∧ ∃ y (RR(y) ∧ CHASE(x,y) ∧ ¬ CATCH(x,y)) → FRUSTRATED(x))*
5. (Conclusion) If all roadrunners say ``beep-beep'', then all coyotes are frustrated.   
   *(∀ x (RR(x) → BEEP(x)) → (∀ y (COYOTE(y) → FRUSTRATED(y)))*

( ( (not (Coyote x)) (RR (f x)) )

( (not (Coyote x)) (Chase x (f x)) )

( (not (RR x)) (not (Beep x)) (Smart x) )

( (not (Coyote x)) (not (RR y)) (not (Smart y)) (not (Catch x y)) )

( (not (Coyote x)) (not (RR y)) (not (Chase x y)) (Catch x y)

(Frustrated x) )

( (not (RR x)) (Beep x) )

( (Coyote (a)) )

( (not (Frustrated (a))) ) )

# MCQs & answers

1. What is the extraction of the meaning of utterance?

(a) Syntactic

(b) Semantic

(c) Pragmatic

(d) None of the above

2. What is the process of associating an FOL

expression with a phrase?

(a) Interpretation

(b) Augmented reality

(c) Semantic interpretation

(d) Augmented interpretation

3. What is meant by compositional semantics?

(a) Determining the meaning

(b) Logical connectives

(c) Semantics

(d) None of the mentioned

4. Which is used to mediate between syntax and semantics?

(a) Form

(b) Intermediate form

(c) Grammar

(d) All of the above

5. What is the process of capturing inference process as a single inference rule?

(a) Ponens

(b) Clauses

(c) Generalised modus ponens

(d) Variables

6. Which process makes different logical expression looks identical?

(a) Lifting

(b) Unification

(c) Inference process

(d) None of the mentioned

7. What among the following could the universal instantiation of \_\_\_\_\_\_\_\_\_\_\_.

For all *x* King(*x*) ^ Greedy(*x*) => Evil(*x*)

(a) King(John) ^ Greedy(John) => Evil(John)

(b) King(y) ^ Greedy(*y*) => Evil(*y*)

(c) King(Richard) ^ Greedy(Richard) => Evil

(Richard)

(d) All of the above

8. What are you predicating by the logic: ۷x: €y: loyalto(x, y).

(a) Everyone is loyal to some one

(b) Everyone is loyal to all

(c) Everyone is not loyal to someone

(d) Everyone is loyal

(e) Everyone is not loyal

9. Which is true for Decision theory?

(a) Decision Theory = Probability theory + utility theory

(b) Decision Theory = Inference theory + utility theory

(c) Decision Theory = Uncertainty + utility theory

(d) Decision Theory = Probability theory + preference

(e) Decision Theory = Probability theory + inference

Answers

1. (b) 2. (c) 3. (a) 4. (b) 5. (c) 6. (b) 7. (d) 8. (a) 9. (a)