Unification

Example :

1. All dogs are animals

   $\forall X\ (dog(X) \Rightarrow animal(X))$.

2. Tommy is a dog

   dog(Tommy).

3. Modus ponens and {Tommy/X} gives

   animal(Tommy).

4. All animals will die

   $\forall Y\ (animal(Y) \Rightarrow die(Y))$.

5. Modus ponens and {Tommy/Y}

   die(Tommy)

Convert predicates to clause form:

Predicate To clause form

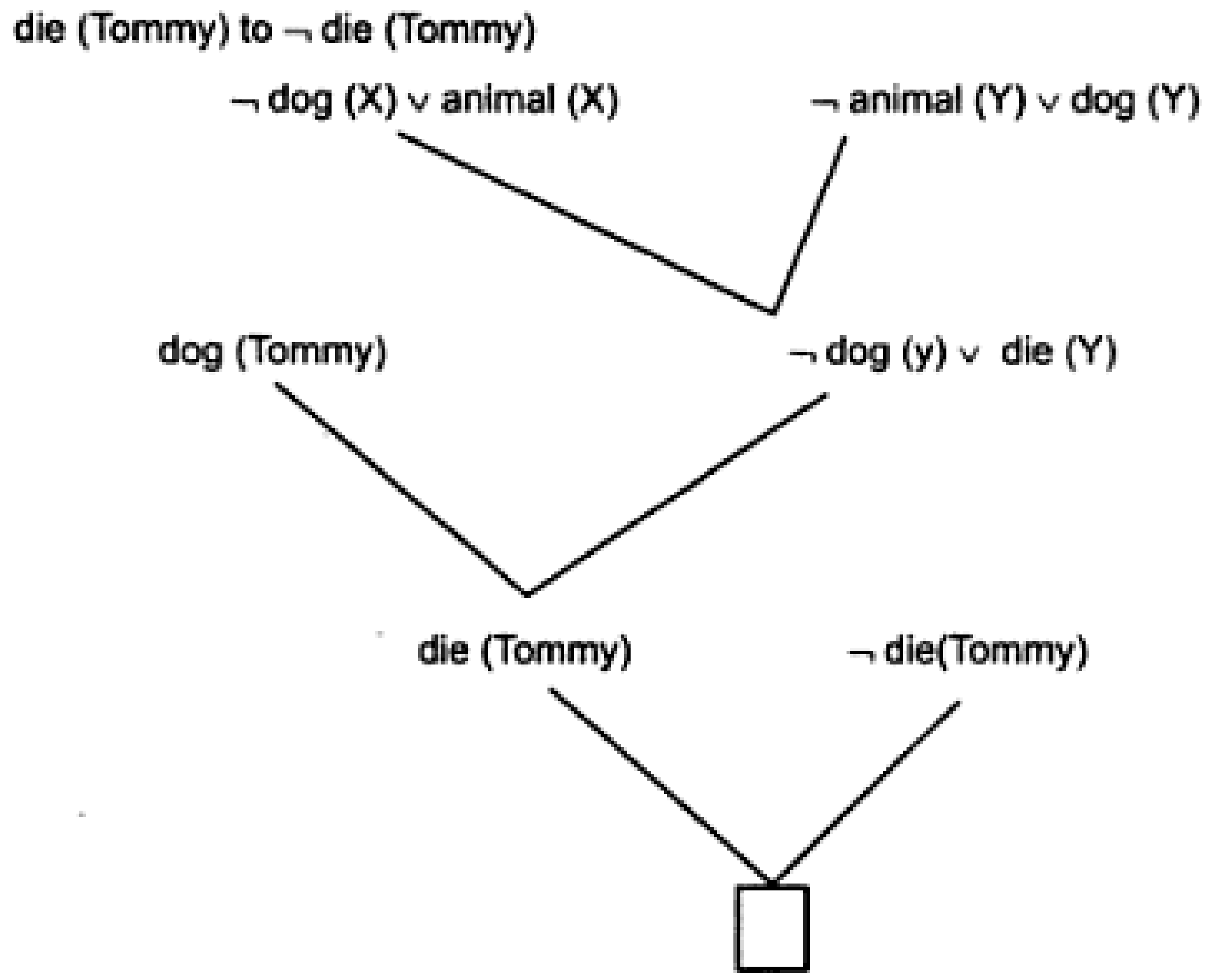
$\forall X\ (dog(X) \Rightarrow animal(X))$ $\neg dog(X) \vee animal(X)$

dog(Tommy)

$\forall Y\ (animal(Y) \Rightarrow die(Y))$ $\neg animal(Y) \vee die(Y)$

Negate the conclusion

die (Tommy) to ¬ die(Tommy)



Unification

To carry out resolution on wff's in FOPL, we need to carry out one final stage, which is to make **substitutions.**

For example, if we had the following set of clauses:

$\{(P(w,x)), (\neg P(y,z))\}$

These clauses can be resolved by making a **substitution.**

We will replace *w* with *y* and *x* with *z*, to result in the following clauses:

$\{(P(y,z)), (\neg P(y,z))\}$

These can now clearly be resolved to give falsum.

The substitution that was made here can be written as

$\{y/w, z/x\}$

with more complex clauses, substitution can be harder

A formal process exists for determining how to make these substitutions, which of course means the process can be automated.

In general, we use the symbol σ to indicate a substitution, and we can write

$\sigma = \{y/w, z/x\}$

$A = P(w,x)$

$B = \neg P(y,z)$

$A\sigma = P(y,z)$

$B\sigma = \neg P(y,z)$

Unification Algorithm

The algorithm for performing the unification is as follows:

Consider X and Y as literals to be unified. The steps of algorithm will be as follows:

1. if X and Y are identical then return nil.
2. else, if P is a variable, then if X occurs in Y then return FAIL, else return (Y/X)
3. if Y is a variable and if Y occurs in X, then return FAIL else return (X/Y).
4. else return FAIL.
5. if the initial predicate symbols in X and Y are not identical, then return FAIL.
6. if the X and Y have different number of arguments then return FAIL.
7. set *substitution* list to nil. (This list contains the substitutions made for performing the unification).
8. for i = 1 to number of arguments in X:
   - 8.1 call unify with arguments of X and the ith argument of Y , putting result in S.
   - 8.2 if S contains FAIL then return (FAIL).
   - 8.3 if S is not equal to nil then
   - 8.4 Apply S to remainder of both X and Y.
   - 8.5 *Substitution* := APPEND (S, *substitution*).
9. return *Substitution*.

Horn Clauses

A **Horn clause**, or **Horn sentence**, is a clause that has, at most, one positive literal. Hence, the following Horn clause takes the following form:

$$A \lor \neg B \lor \neg C \lor \neg D \lor \neg E \ldots$$

where *A*, *B*, *C*, *D*, *E*, and so on are positive literals.

This Horn clause can also be written as an implication:

$$B \land C \land D \land E \rightarrow A$$

Horn clauses can take three forms.

- ✓ **rule relation**
- ✓ **fact:**
- ✓ **goal**, or a **headless clause**

The type we have seen above, where there is one positive literal, and one or more negative literals is called a **rule relation**

A clause with no negative literals is called a **fact:**

A :–

Finally, a clause with no positive literal is called a **goal**, or a **headless clause:**

:– B, C, D, E

Unification Example

Let us find the mgu for the following set of clauses:

$S_0 = \{P(a, x, y, z), P(x, y, z, w)\}$

(*a* is a constant, and *x*, *y*, *z* are variables).

We initialize $\sigma_0 = \{\}$ and $i = 0$.

Now we proceed as follows:

$D_0 = \{a,x\}$

$\sigma_1 = \sigma_o \circ \{a/x\} = \{a/x\}$

$S_1 = S_o \{a/x\} = \{P(a, a, y, z), P(a, y, z, w)\}$

$D_1 = \{a,y\}$

$\sigma_2 = \{a/x\} \circ \{a/y\} = \{a/x, a/y\}$

$S_2 = S_1 \{a/x, a/y\} = \{P(a, a, a, z), P(a, a, z, w)\}$

$D_2 = \{a, z\}$

$\sigma_3 = \{a/x, a/y\} \circ \{a/z\} = \{a/x, a/y, a/z\}$

$S_3 = S_2 \{a/x, a/y, a/z\} = \{P(a, a, a, a), P(a, a, a, w)\}$

$D_3 = \{a, w\}$

$\sigma_4 = \{a/x, a/y, a/z\} \circ \{a/w\} = \{a/x, a/y, a/z, a/w\}$

$S_4 = S_3 \{a/x, a/y, a/z, a/w\} = \{P(a, a, a, a), P(a, a, a, a)\}$

$\quad = \{P(a, a, a, a)\}$

Now we stop because S_4 has just one element, and we have found a mgu, σ_4, which is $\{a/x, a/y, a/z, a/w\}$.

Unification

The process of finding the substitutions needed to make two predicate calculus expressions match is called Unification. Unification is a "pattern matching" procedure that takes two atomic sentences, called **literals**, as input, and returns "failure" if they do not match and a substitution list, `Theta`, if they do match.

i.e. unify(p,q) = Theta means subst (Theta, p) = subst(Theta, q) for two atomic sentences `p` and `q`. Theta is called the **most general unifier (mgu)** .All variables in the given two literals are implicitly universally quantified. To make literals match, replace (universally-quantified) variables by terms

Example:	{play(X,a,dance(Y), play(Shree, a, dance(Z))}
Substitution	{Shree/X} yields
	{play(Shree,a,dance(Y), play(Shree, a, dance(Z))}
Substitution:	{Shree/X, Z/Y} yields
	{play(Shree,a,dance(Z), play(Shree, a, dance(Z))}

here 'Shree/X' indicates that 'Shree' is substituted for the variable X or the variable X is said to be bound to the value 'Shree'

Unification rules

- A variable may be replaced by a constant: c/X
- A variable may be replaced by a variable: Y/X
- A variable may be replaced by a function expression as long as the function expression does not contain the variable: p(Y)/X
- Once a variable has been bound, future unification and inferences must take this substitution into account

Function unify

Function unify (E1,E2) computes the unifying substitutions between two predicate calculus expressions automatically.

A sentence is presented as a list with the predicate or function symbol as the first element followed by its arguments

Examples:

p(a,b) ⇔ (p a b)

p(f(a),g(X,Y)) ⇔ (p (f a) (g X Y))

equal(eve,mother(cain)) ⇔ (equal eve (mother cain))

$p(X) \wedge q(Y)$ ⇔ ((p X) $\wedge$ (q Y))

Unify is a linear time algorithm that returns the **most general unifier (mgu)**, i.e., a shortest length substitution list that makes the two literals match. (In general, there is not a unique minimum length substitution list, but unify returns one of those of minimum length.) A variable can never be replaced by a term containing that variable. For example, `x/f(x)` is illegal.

Questions

Q1. *From the given set of facts proof that:*

1. *Anyone passing his history exams and winning the lottery is happy.*
2. *But anyone who studies or is lucky can pass all his exams.*
3. *Shree did not study but he is lucky.*
4. *Anyone who is lucky wins the lottery.*
5. *Is Shree happy?*

Q.2

Consider the following axioms:Convert to clause form

1. *Anyone who buys carrots by the bushel owns either a rabbit or a grocery store.*
2. *Every dog chases some rabbit.*
3. *Mary buys carrots by the bushel.*
4. *Anyone who owns a rabbit hates anything that chases any rabbit.*
5. *John owns a dog.*
6. *Someone who hates something owned by another person will not date that person.*
7. *Prove that if Mary does not own a grocery store, she will not date John.*

Q3 *Consider the following predicates and proof::-*

a) John likes all kinds of food.

b) Apples are food.

c) Chicken is food.

d) Anything any are eats and is not killed by is food.

e) Bill eats peanuts and is still alive.

f) Sue eats everything Bill eats.

Solve

1) Translate to predicate logic.

2) P. T. John likes peanuts.

3) Convert the formulae into clause form.

4) P.T. John likes peanuts using resolution.

5) Use resolution to answer what food Sue likes.

Q4.

Consider foll. facts : -

a) The members of the bridge club are Joe, Sally, Bill and Allen.

b) Joe is married to Sally.

c) The spouse of every married person in the club is also in the club.

d) The last meeting of the club was in Joe's house.

Represent above facts in predicate logic

A) P. T. Allen is not married.

Q. 5 *Prove what course would Steve like*

Assume the foll facts

- *Steve only likes easy courses.*
- *Science courses are hard.*
- *All the courses in the basketweaving department are easy.*
- *BK301 is a basketweaving course.*

Use resolution to prove "what course would Steve Like.

Q.6 *Principle of resolution of predicate logic.*

Q. 7 *Consider the following axioms:*

1. *Anyone whom Mary loves is a football star.*
2. *Any student who does not pass does not play.*
3. *John is a student.*
4. *Any student who does not study does not pass.*
5. *Anyone who does not play is not a football star.*
6. *Prove that if John does not study, then Mary does not love John.*

Q. 8 *What are the strategies used to speed up resolution proof?*

Q.9 *Convert following sentence into WFF "The spouse of every married person in the club is also in the club."*

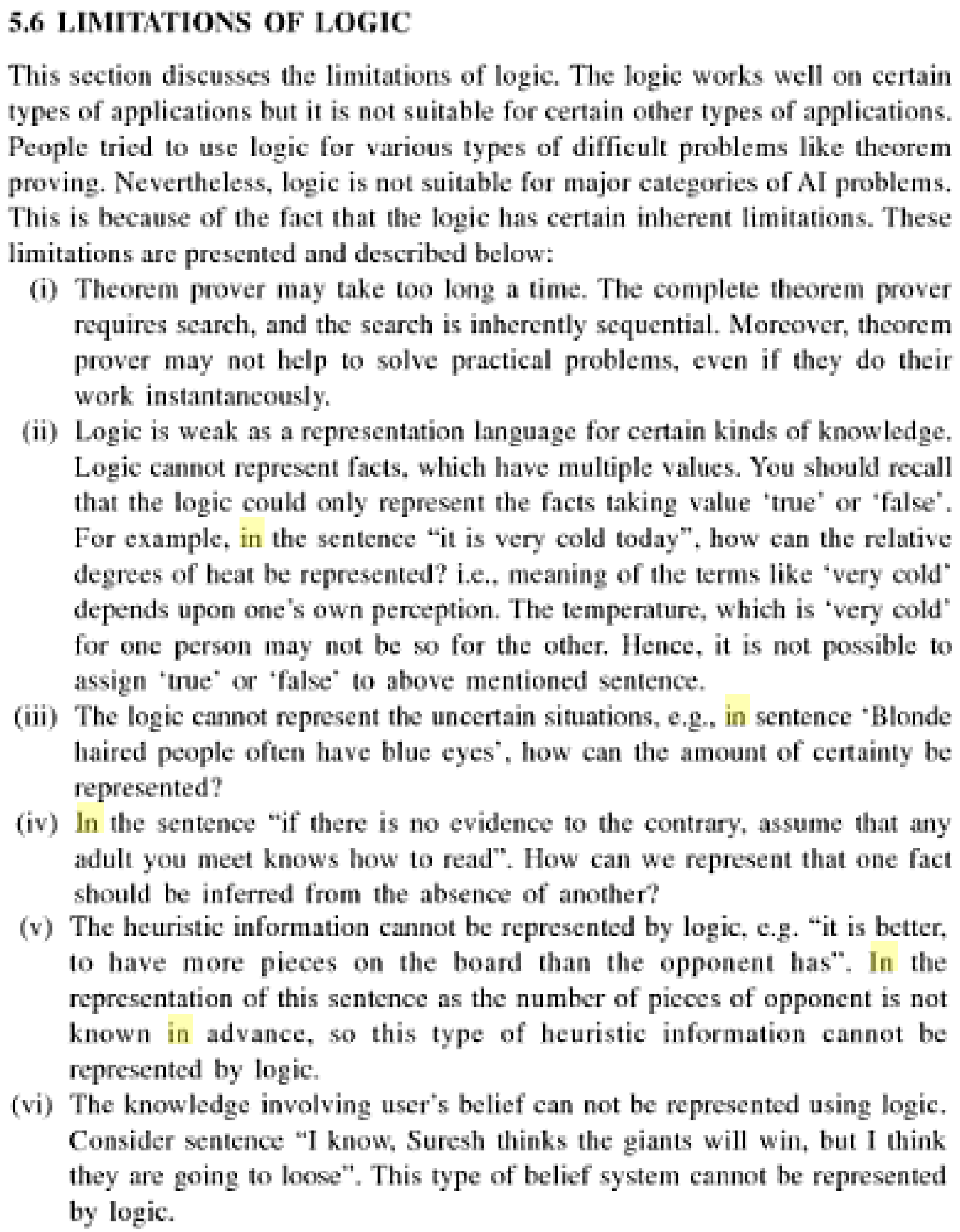
Q.10 *Frame WFF FOR*

There are 3 blocks. A block can be placed over another block if doesn't have a block over it. A block can be placed over the table.

Q.11 *Represent following facts in predicate logic in such a way that the conclusion cannot be derived.*

FACTS: Socrates is a man. Men are widely distributed over the earth.

Conclusion: Socrates is widely distributed over the earth.



5.6 LIMITATIONS OF LOGIC

This section discusses the limitations of logic. The logic works well on certain types of applications but it is not suitable for certain other types of applications. People tried to use logic for various types of difficult problems like theorem proving. Nevertheless, logic is not suitable for major categories of AI problems. This is because of the fact that the logic has certain inherent limitations. These limitations are presented and described below:

(i) Theorem prover may take too long a time. The complete theorem prover requires search, and the search is inherently sequential. Moreover, theorem prover may not help to solve practical problems, even if they do their work instantaneously.

(ii) Logic is weak as a representation language for certain kinds of knowledge. Logic cannot represent facts, which have multiple values. You should recall that the logic could only represent the facts taking value 'true' or 'false'. For example, in the sentence "it is very cold today", how can the relative degrees of heat be represented? i.e., meaning of the terms like 'very cold' depends upon one's own perception. The temperature, which is 'very cold' for one person may not be so for the other. Hence, it is not possible to assign 'true' or 'false' to above mentioned sentence.

(iii) The logic cannot represent the uncertain situations, e.g., in sentence 'Blonde haired people often have blue eyes', how can the amount of certainty be represented?

(iv) In the sentence "if there is no evidence to the contrary, assume that any adult you meet knows how to read". How can we represent that one fact should be inferred from the absence of another?

(v) The heuristic information cannot be represented by logic, e.g. "it is better, to have more pieces on the board than the opponent has". In the representation of this sentence as the number of pieces of opponent is not known in advance, so this type of heuristic information cannot be represented by logic.

(vi) The knowledge involving user's belief can not be represented using logic. Consider sentence "I know, Suresh thinks the giants will win, but I think they are going to loose". This type of belief system cannot be represented by logic.