PROGRAMME DESIGN COMMITTEE

Prof. (Retd.) S.K. Gupta , IIT, Delhi
Prof. Ela Kumar, IGDTUW, Delhi
Prof. T.V. Vijay Kumar JNU, New Delhi
Prof. Gayatri Dhingra, GVMITM, Sonipat
Mr. Milind Mahajan,. Impressico Business Solutions, New Delhi
Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU
Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU
Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU
Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

COURSE DESIGN COMMITTEE

Prof. T.V. Vijay Kumar, JNU, New Delhi
Prof. S.Balasundaram, JNU, New Delhi
Prof D.P. Vidyarthi, JNU, New Delhi
Prof. Anjana Gosain, USICT, GGSIPU, New Delhi
Dr. Ayesha Choudhary, JNU, New Delhi
Sh. Shashi Bhushan Sharma, Associate Professor, SOCIS, IGNOU
Sh. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. P. Venkata Suresh, Associate Professor, SOCIS, IGNOU
Dr. V.V. Subrahmanyam, Associate Professor, SOCIS, IGNOU
Sh. M.P. Mishra, Assistant Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU

SOCIS FACULTY

Prof. P. Venkata Suresh, Director, SOCIS, IGNOU
Prof. V.V. Subrahmanyam, SOCIS, IGNOU
Dr. Akshay Kumar, Associate Professor, SOCIS, IGNOU
Dr. Naveen Kumar, Associate Professor, SOCIS, IGNOU (on EOL)
Dr. M.P. Mishra, Associate Professor, SOCIS, IGNOU
Dr. Sudhansh Sharma, Assistant Professor, SOCIS, IGNOU
Dr. Manish Kumar, Assistant Professor, SOCIS, IGNOU

PREPARATION TEAM

Dr.Sudhansh Sharma, (Writer- Unit 5, 6)
Asst.Professor SOCIS, IGNOU
(Writer- Unit 5, 6)-(Partially Adapted from MCSE003 Artificial Intelligence & Knowledge Management)

Dr. Manish Kumar, Assistant Professor
SOCIS, IGNOU (Writer- Unit 7)

Dr.Sudhansh Sharma, (Writer- Unit 8)
Asst.Professor SOCIS, IGNOU
(Writer- Unit 8)-(Partially Adapted from MCSE003 Artificial Intelligence & Knowledge Management)

Prof Ela Kumar (Content Editor)
Department of Computers & Engg. IGDTUW, Delhi

Prof.Parmod Kumar (Language Editor)
SOH, IGNOU, New Delhi

Course Coordinator: Dr.Sudhansh Sharma,

Print Production

Sh Sanjay Aggarwal,Assistant Registrar, MPDD

, 2022

ISBN-

Further information on the Indira Gandhi National Open University courses may be obtained from the University's office at Maidan Garhi, New Delhi-110068.
Printed and published on behalf of the Indira Gandhi National Open University, New Delhi by MPDD, IGNOU.

UNIT 5 FIRST ORDER PREDICATE LOGIC

Structure **Page Nos.**

5.0 INTRODUCTION

In the previous unit, we discussed how propositional logic helps us in solving problems. However, one of the major problems with propositional logic is that, sometimes, it is unable to capture even elementary type of reasoning or argument as represented by the following statements:

Every man is mortal.

Raman is a man.

Hence, he is mortal.

The above reasoning is intuitively correct. However, if we attempt to simulate the reasoning through Propositional Logic and further, for this purpose, we use symbols P, Q and R to denote the statements given above as:

P: Every man is mortal,

Q: Raman is a man,

R: Raman is mortal.

Once, the statements in the argument in English are symbolised to apply tools of propositional logic, we just have three symbols P, Q and R available with us and apparently no link or connection to the original statements or to each other. The connections, which would have helped in solving the problem become invisible. In Propositional Logic, there is no way, to conclude the *symbol* R from the *symbols* P and Q. However, as we mentioned earlier, even in a natural language, the conclusion of the *statement* denoted by R from *the statements* denoted by P and Q is obvious. Therefore, we search for some **symbolic** system of reasoning that helps us in discussing *argument forms* of the above-mentioned type, in addition to those forms which can be discussed within the framework of propositional logic. **First Order Predicate Logic (FOPL)** is the most well-known symbolic system for the pourpose.

The symbolic system of FOPL treats an atomic statement *not as an indivisible unit*. Rather, FOPL not only treats an atomic statement divisible into subject and predicate but even further deeper structures of an atomic statement are considered in order to handle larger class of arguments. How and to what extent

FOPL symbolizes and establishes *validity/invalidity* and *consistency/inconsistency* of *arguments* is the subject matter of this unit.

5.1 OBJECTIVES

After studying this unit, you should be able to:

- explain why FOPL is required over and above PL;
- define, and give appropriate examples for, each of the new concepts required for FOPL including those of quantifier, variable, constant, term, free and bound occurrences of variables, closed and open wff;
- check consistency/validity, if any, of closed formulas;
- reduce a given formula of FOPL to normal forms: Prenex Normal Form (PNF) and (Skolem) Standard Form, and conversion to the clausal form
- use the tools and techniques of FOPL, developed in the unit, to solve problems requiring logical reasoning
- Perform unification and resolution mechanism.

5.2 SYNTAX OF FIRST ORDER PREDICATE LOGIC

We learned about the concept of propositions in Artificial intelligence, in Unit 4 of Block 1. Now it's time to understand the difference between the Proposition and the Predicate (also known as propositional function). In short, a proposition is a specialized statement whereas Predicate is a generalized statement. To be more specific the propositions uses the logical connectives only and the predicates uses logical connectives and quantifiers (universal and existential), both.

Note : ∃ is the symbol used for the Existential quantifier and ∀ is used for the Universal quantifier.

Let's understand the difference through some more detail, as given below.

A propositional function, or a **predicate,** in a variable x is a sentence p(x) involving x that becomes a proposition when we give x a definite value from the set of values it can take. We usually denote such functions by p(x), q(x), etc. The set of values x can take is called the universe of discourse.

So, if p(x) is '$x > 5$', then p(x) is not a proposition. But when we give x particular values, say $x = 6$ or $x = 0$, then we get propositions. Here, p(6) is a true proposition and p(0) is a false proposition.

Similarly, if q(x) is 'x has gone to Patna.', then replacing x by 'Taj Mahal' gives us a false proposition.

Note that a predicate is usually not a proposition. But, of course, every proposition is a prepositional function in the same way that every real number is a real-valued function, namely, the constant function.

Now, can all sentences be written in symbolic from by using only the logical connectives? What about sentences like 'x is prime and x + 1 is prime for some x.'? How would you symbolize the phrase 'for some x', which we can rephrase as 'there exists an x'? You must have come across this term often while studying mathematics. **We use the symbol '∃' to denote this quantifier, 'there exists'.** The way we use it is, for instance, to rewrite 'There is at least one child in the class.' as'(∃ x in U)p(x)',

where p(x) is the sentence 'x is in the class.' and U is the set of all children.

Now suppose we take the negative of the proposition we have just stated. Wouldn't it be 'There is no child in the class.'? We could symbolize this as 'for all x in U, q(x)' where x ranges over all children and q(x) denotes the sentence 'x is not in the class.', i.e., $q(x) \equiv \sim p(x)$.

We have a **mathematical symbol for the quantifier 'for all', which is '$\forall$'.** So the proposition above can be written as

'$(\forall x \in U)q(x)$', or '$q(x), \forall x \in U$'.

An example of the use of the existential quantifier is the true statement.

$(\exists x \in \mathbf{R})(x + 1 > 0)$, which is read as 'There exists an x in **R** for which x + 1 > 0.'.

Another example is the false statement

$(\exists x \in \mathbf{N})(x - \frac{1}{2} = 0)$, which is read as 'There exists an x in **N** for which $x - \frac{1}{2} = 0$.'.

An example of the use of the universal quantifier is $(\forall x \notin \mathbf{N})(x^2 > x)$, which is read as 'for every x not in **N**, $x^2 > x$.'. Of course, this is a false statement, because there is at least one $x \notin \mathbf{N}$, $x \in \mathbf{R}$, for which it is false.

As you have already read in the example of a child in the class,

$(\forall x \in U)p(x)$ is logically equivalent to $\sim (\exists x \in U)(\sim p(x))$. Therefore,

$\sim(\forall x \in U)p(x) \equiv \sim\sim (\exists x \in U)(\sim p(x)) \equiv (\exists x \in U)(\sim p(x))$.

This is one of the rules for negation that relate $\forall$ and $\exists$. The two rules are

$\sim (\forall \mathbf{x} \in \mathbf{U})\mathbf{p}(\mathbf{x}) \equiv (\exists x \in \mathbf{U})(\sim \mathbf{p}(\mathbf{x}))$, and

$\sim (\exists \mathbf{x} \in \mathbf{U})\mathbf{p}(\mathbf{x}) \equiv (\forall x \in \mathbf{U})(\sim \mathbf{p}(\mathbf{x}))$

Where U is the set of values that x can take.

5.3 INTERPRETATIONS IN FOPL

In order to have a glimpse at how FOPL extends propositional logic, let us again discuss the earlier argument.

Every man is mortal. Raman is a man.

Hence, he is mortal.

In order to derive the validity of above simple argument, instead of looking at an atomic statement as indivisible, to begin with, we divide each statement into *subject* and *predicate*. The two predicates which occur in the above argument are:

'*is mortal'* and *'is man'*.

Let us use the notation

IL: *is_mortal* and

IN: *is_man*.

In view of the notation, the argument on para-phrasing becomes:
For all x, if IN (x) then IL (x).

IN (Raman).

Hence, IL (RAMAN)

More generally, relations of the form *greater-than (x, y)* denoting the phrase 'x is greater than y', *is_brother_ of* (x, y) denoting '*x is brother of y,*' *Between (x, y, z)* denoting the phrase that '*x lies between y and z*', and *is_tall (x)* denoting '*x is tall*' are some **examples of predicates**. The variables x, y, z etc which appear in a predicate are called **parameters** of the predicate.

The parameters may be given some appropriate values such that after substitution of appropriate value from all possible values of each of the variables, the predicates become *statements*, for each of which we can say whether it is 'True' or it is 'False'.

For example, for the predicate *greater-than (x, y),* if x is given value 3 then we obtain *greater-than (3, y),* for which still it is not possible to tell whether it is True or False. Hence, '*greater-than (3, y)*' is also a predicate. Further, if the variable y is given value 5 then we get *greater (3, 5)* which , as we known, is False. Hence, it is possible to give its Truth-value, which is *False* in this case. Thus, from the *predicate greater-than (x, y)*, we get the *statement greater-than (3, 5)* by assigning values 3 to the variable x and 5 to the variable y. These values 3 and 5 are called parametric values or *arguments* of the predicate *greater-than*.

(Please note 'argument of a function/predicate' is a mathematical concept, different from logical argument)

Similarly, we can represent the phrase *x likes y* by the *predicate LIKE (x, y)*. Then *Ram likes Mohan* can be represented by the statement *LIKE (RAM, MOHAN)*.

Also *function symbols* can be used in the first-order logic. For example, we can use *product (x, y)* to denote $x * y$ and *father (x)* to mean the *'father of x'*. The statement: *Mohan's father loves Mohan* can be symbolised as *LOVE (father (Mohan), Mohan)*. Thus, we need not know name of father of Mohan and still we can talk about him. A function serves such a role.

We may note that LIKE (Ram, Mohan) and LOVE (father (Mohan),Mohan) are atoms or atomic statements of PL, in the sense that, one can associate a truth-value *True* or *False* with each of these, and each of these does not involve a logical operator like $\sim, \wedge, \vee, \rightarrow$ or $\leftrightarrow$.

Summarizing in the above discussion, *LIKE (Ram, Mohan)* and *LOVE (father (Mohan) Mohan)* are **atoms;** where as GREATER, LOVE and LIKE are **predicate symbols;** x and y are **variables and 3**, *Ram* and *Mohan* are **constants;** and *father* and *product* are **function symbols**.

From the above discussion we learned the following concepts of symbols.

i) ***Individual symbols or*** *constant symbols*: These are usually names of objects, such as Ram, Mohan, numbers like 3, 5 etc.
ii) **Variable symbols**: These are usually lowercase unsubscripted or subscripted letters, like x, y, z, x_3.

iii) **Function symbols:** These are usually lowercase letters like f, g, h,….or strings of lowercase letters such as *father* and *product*.
iv) **Predicate symbols***:* These are usually uppercase letters like P, Q, R,….or strings of lowercase letters such as *greater-than*, *is_tall* etc.

A function symbol or predicate symbol takes a fixed number of arguments. If a *function symbol f* takes *n* arguments, *f* is called an *n-place function symbol*. Similarly, if a predicate symbol Q takes *m* arguments, P is called an m-*place predicate symbol*. For example, father is a one-place *function* symbol, and GREATER and LIKE are two-place *predicate* symbols. However, *father-of* in father_of (x, y) is a, *two place predicate* symbol.

The symbolic representation of an argument of a function or a predicate is called a *term* where a **term** is defined recursively as follows:

i) A variable is a term.
ii) A constant is a term.
iii) If *f* is an n-place function symbol, and $t_1 \ldots t_n$ are terms, then $f(t_1,\ldots,t_n)$ is a term.
iv) Any term can be generated only by the application of the rules given above.

For example: Since, *y* and 3 are both terms and *plus* is a two-place function symbol, *plus (y, 3)* is a term according to the above definition.

Furthermore, we can see that *plus (plus (y, 3), y)* and *father (father (Mohan))* are also terms; the former denotes *(y + 3) + y* and the later denotes *grandfather of Mohan*.

A predicate can be thought of as a function that maps a list of constant arguments to T or F. For example, GREATER is a predicate with GREATER (5, 2) as T, but GREATER (1, 3) as F.

We already know that in PL, an atom or atomic statement is an indivisible unit for representing and validating arguments. Atoms in PL are denoted generally by symbols like P, Q, and R etc. But in FOPL,

Definition: An Atom is

(i) either an atom of Propositional Logic, or
(ii) is obtained from an n-place predicate symbol P, and terms $t_1,\ldots t_n$ so that $P\ (t_1,\ldots,t_n)$ is an atom.

Once, the atoms are defined, by using the logical connectives defined in Propositional Logic, and assuming having similar meaning in FOPL, we can build complex formulas of FOPL. Two special symbol ∀ and ∃ are used to denote qualifications in FOPL. The symbols ∀ and ∃ are called, respectively, the *universal* quantifier and *existential* quantifier. For a variable x, (∀x) is read as *for all x,* and (∃x) is read as *there exists an x.* Next, we consider some examples to illustrate the concepts discussed above.

In order to symbolize the following statements:

i) There exists a number that is rational.
ii) Every rational number is a real number
iii) For every number *x*, there exists a number *y*, which is greater than *x*.

let us denote *x is a rational number* by Q(x), *x is a real number* by R(x), and *x is less than y* by *LESS(x, y)*. Then the above statements may be symbolized respectively, as

(i) $(\exists x)\ Q(x)$

(ii) $(\forall x)\ (Q(x) \rightarrow R\ (x))$

(iii) $(\forall x)\ (\exists y)$ LESS(x, y).

Each of the expressions (i), (ii), and (iii) is called a **formula** or a well-formed formula or **wff.**

5.4 SEMATICS OF QUANTIFIERS

To understand the semantics of quantifiers we need to first understand the difference between the Proposition and the Predicate(also known as propositional function). In short, a proposition is a specialized statement whereas Predicate is a generalized statement. To be more specific the propositions uses the logical connectives only and the predicates uses logical connectives and quantifiers (universal and existential), both.

Note : $\exists$ is the symbol used for the Existential quantifier and $\forall$ is used for the Universal quantifier.

Let's understand the difference through some more detail, as given below.

A propositional function, or a **predicate,** in a variable x is a sentence p(x) involving x that becomes a proposition when we give x a definite value from the set of values it can take. We usually denote such functions by p(x), q(x), etc. The set of values x can take is called the universe of discourse.

So, if p(x) is '$x > 5$', then p(x) is not a proposition. But when we give x particular values, say $x = 6$ or $x = 0$, then we get propositions. Here, p(6) is a true proposition and p(0) is a false proposition.

Similarly, if q(x) is 'x has gone to Patna.', then replacing x by 'Taj Mahal' gives us a false proposition.

Note that a predicate is usually not a proposition. But, of course, every proposition is a prepositional function in the same way that every real number is a real-valued function, namely, the constant function.

Now, can all sentences be written in symbolic from by using only the logical connectives? What about sentences like 'x is prime and x + 1 is prime for some x.'? How would you symbolize the phrase 'for some x', which we can rephrase as 'there exists an x'? You must have come across this term often while studying mathematics. **We use the symbol '$\exists$' to denote this quantifier, 'there exists'.** The way we use it is, for instance, to rewrite 'There is at least one child in the class.' as'($\exists$ x in U)p(x)',

where p(x) is the sentence 'x is in the class.' and U is the set of all children.

Now suppose we take the negative of the proposition we have just stated. Wouldn't it be 'There is no child in the class.'? We could symbolize this as 'for all x in U, q(x)' where x ranges over all children and q(x) denotes the sentence 'x is not in the class.', i.e., $q(x) \equiv \sim p(x)$.

We have a **mathematical symbol for the quantifier 'for all', which is '$\forall$'.** So the proposition above can be written as

'$(\forall\ x \in U)q(x)$', or '$q(x), \forall\ x \in U$'.

An example of the use of the existential quantifier is the true statement.

$(\exists\ x \in \mathbf{R})\ (x + 1 > 0)$, which is read as 'There exists an x in **R** for which x + 1 > 0.'.

Another example is the false statement

$(\exists\ x \in \mathbf{N})\ (x - \frac{1}{2} = 0)$, which is read as 'There exists an x in **N** for which $x - \frac{1}{2} = 0$.'.

An example of the use of the universal quantifier is $(\forall\ x \notin \mathbf{N})\ (x^2 > x)$, which is read as 'for every x not in **N**, $x^2 > x$.'. Of course, this is a false statement, because there is at least one $x \notin \mathbf{N}$, $x \in \mathbf{R}$, for which it is false.

As you have already read in the example of a child in the class,

$(\forall\ x \in U)p(x)$ is logically equivalent to $\sim(\exists\ x \in U)\ (\sim p(x))$. Therefore,

$\sim(\forall\ x \in U)p(x) \equiv \sim\sim(\exists\ x \in U)\ (\sim p(x)) \equiv (\exists\ x \in U)\ (\sim p(x))$.

This is one of the rules for negation that relate $\forall$ and $\exists$. The two rules are

$\sim(\forall\ \mathbf{x} \in \mathbf{U})\mathbf{p}(\mathbf{x}) \equiv (\exists\ x \in \mathbf{U})\ (\sim \mathbf{p}(\mathbf{x}))$, and

$\sim(\exists\ \mathbf{x} \in \mathbf{U})\mathbf{p}(\mathbf{x}) \equiv (\forall\ x \in \mathbf{U})\ (\sim \mathbf{p}(\mathbf{x}))$

Where U is the set of values that x can take.

Next, we discuss three new concepts, viz **Scope** of occurrence of a quantified variable, Bound occurrence of a quantifier variable or quantifier and *Free occurrence* of a variable.

Before discussion of these concepts, we should know the *difference between a **variable** and **occurrence of a variable** in a quantifier expression*.

The variable *x* has THREE occurrences in the formula

$(\exists x)\ Q(x) \rightarrow P(x, y)$.

Also, the variable *y* has only one occurrence and the variable *z* has zero occurrence in the above formula. Next, we define the three concepts mentioned above.

Scope of an occurrence of a quantifiers is the smallest but complete formula following the quantifier sometimes delimited by pair *f* parentheses. For example, Q(x) is the scope of $(\exists x)$ in the formula

$(\exists x)\ Q(x) \rightarrow P(x, y)$.

But the scope of $(\exists x)$ in the formula: $(\exists x)\ (Q(x) \rightarrow P(x, y))$ is $(Q(x) \rightarrow P(x, y))$.

Further in the formula:

$(\exists x)\ (P(x) \rightarrow Q(x, y)) \wedge (\exists x)\ (P(x) \rightarrow R(x, 3))$,

the scope of **first** occurrence of $(\exists x)$ is the formula $(P(x) \rightarrow Q\ (x, y)$ and the scope of **second** occurrence of $(\exists x)$ is the formula

$(P(x) \rightarrow R(x, 3))$.

As another example, the scope of the only occurrence of the quantifier $(\forall y)$ in

$(\exists x)$ (($P(x) \rightarrow Q(x) \leftrightarrow (\forall y)$ (Q (x) $\rightarrow$ R (y))) is (Q (x) $\rightarrow$ R(y)). But the scope of the only occurrence of the existential variable $(\exists x)$ in the same formula is the formula:

$(P(x) \rightarrow Q(x))$ P $\leftrightarrow (\forall y)$ (Q (x) $\rightarrow$ R(y))

An ***occurrence*** *of a variable in a formula* is **bound** if and only if the occurrence is within the scope of a quantifier employing the variable, or is the occurrence in that quantifier. An occurrence of a variable in a formula is **free** if and only if this occurrence of the variable is not bound.

Thus, in the formula $(\exists x)$ P(x, y) $\rightarrow$ Q (x), there are three occurrences of *x*, out of which first two occurrences of *x* are *bound*, where, the last occurrence of *x is free*, because scope of $(\exists x)$ in the above formula is P(x, y). The only occurrence of *y* in the formula is free. Thus, *x* is both a bound and a free variable in the above formula and y is only a free variable in the formula so far, we talked of an *occurrence of a variable* as free or bound. Now, we talk of (only) a *variable* as free or bound. *A variable is* ***free*** in a formula if at least one occurrence of it is free in the formula. A variable is **bound** in a formula if at least one occurrence of it is bound.

It may be noted that a variable can *be* ***both free and bound*** *in a formula*. In order to further elucidate the concepts of *scope, free and bound occurrences of a variable*, we consider a similar but different formula for the purpose:

$(\exists x)$ (P(x, y) $\rightarrow$ Q(x)).

In this formula, *scope* of the only occurrence of the quantifier $(\exists x)$ is the whole of the rest of the formula, viz. scope of $(\exists x)$ in the given formula is (P(x, y) $\rightarrow$ Q (x))

Also, all three occurrence of variable *x* are bound. The only occurrence of *y* is free.

Remarks: It may be noted that a bound variable *x* is just a **place holder** or a **dummy variable** in the sense that all occurrences of a bound variable *x* may be replaced by another free variable say *y*, which does not occur in the formula. However, once, *x* is replaced by *y* then *y* becomes bound. For example, $(\forall x)$ (f (x)) is the same as $(\forall y)$ f (y). It is something like

$$\int_1^2 x^2 dx = \int_1^2 y^2 dy = \frac{2^3}{3} - \frac{1^3}{3} = \frac{7}{3}$$

Replacing a bound variable *x* by another variable *y* under the restrictions mentioned above is called **Renaming of a variable x**

Having defined an atomic formula of FOPL, next, we consider the definition of a general formula formally in terms of atoms, logical connectives, and quantifiers.

Definition A well-formed formula, **wff** a just or formula in FOPL is defined recursively as follows:

i) An atom or atomic formula is a *wff.*

ii) If E and G are wff, then each of ~ (E), $(E \vee G)$, $(E \wedge G)$, $(E \rightarrow G)$, $(E \leftrightarrow G)$ is a **wff.**

iii) If E is a *wff* and x is a free variable in E, then $(\forall x)E$ is a wff.

iv) A *wff* can be obtained only by applications of (i), (ii), and (iii) given above.

We may drop pairs of parentheses by agreeing that quantifiers have the least scope. For example, $(\exists x)\ P(x, y) \rightarrow Q(x)$ stands for

$((\exists x)\ P(x, y)) \rightarrow Q(x)$

We may note the following two cases of translation:

(i) for all x, P(x) is Q(x) is translated as
$(\forall x)\ (P(x) \rightarrow Q(x))$
(the other possibility $(\forall x)\ P(x) \wedge Q(x)$ *is not valid.)*

(ii) for some x, P(x) is Q (x) is translated as $(\exists x)\ P(x) \wedge Q(x)$
(the other possibility $(\exists x)\ P(x) \rightarrow Q(x)$ *is not valid)*

Example
Translate the statement: *Every man is mortal. Raman is a man. Therefore, Raman is mortal.*

As discussed earlier, let us denote "*x is a man*" by MAN (x), and "*x is mortal*" by MORTAL(x). Then "*every man is mortal*" can be represented by

$$(\forall x)\ (MAN(x) \rightarrow MORTAL(x)),$$

"Raman is a man" by

$$MORTAL(Raman).$$

The whole argument can now be represented by

$(\forall x)\ (MAN(x) \rightarrow MORTAL(x)) \wedge MAN\ (Roman) \rightarrow MORTAL\ (Roman).$

as a single statement.

In order to further explain symbolisation let us recall the axioms of natural numbers:

(1) For every number, there is one and only one immediate successor,

(2) There is no number for which 0 is the immediate successor.

(3) For every number other than 0, there is one and only one immediate predecessor.

Let the *immediate successor* and *predecessor of x*, respectively be denoted by f(x) and g(x).

Let E *(x, y)* denote *x is equal to y*. Then the axioms of natural numbers are represented respectively by the formulas:

(i) $(\forall x)\ (\exists y)\ (E(y, f(x)) \wedge (\forall z)\ (E(z, f(x)) \rightarrow E(y, z)))$

(ii) $\sim ((\exists x)\ E(0, f(x)))$ and

(iii) $(\forall x)\ (\sim E(x, 0) \rightarrow ((y)\exists,\ g(x)) \wedge (\forall z)\ (E(z, g(x)) \rightarrow E(y, z)))))$.

From the semantics (for meaning or interpretation) point of view, the **wff of FOPL** may be divided into two categories, each consisting of

(i) wffs, in each of which, **all** occurrences of variables are **bound**.
(ii) wffs, in each of which, at **least one** occurrence of a variable is **free**.

The wffs of FOPL in which there is no occurrence of a free variable, are like *wffs* of PL in the sense that we can call each of the wffs as **True, False, consistent, inconsistent, valid, invalid etc**. Each such a formula is called **closed formula**. However, when a wff involves a free occurrence, then it is not possible to call such a wff as True, False etc. **Each of such a formula is called an open formula**.

For example: Each of the formulas: greater (x, y), greater (x, 3), $(\forall y)$ greater (x, y) has one free occurrence of variable x. Hence, each is an **open** formula.
Each of the formulas: $(\forall x)\ (\exists y)$ greater (x, y), $(\forall y)$ greater (y, 1), greater (9, 2), does not have free occurrence of any variable. Therefore each of these formulas is a closed formula.

Next we discuss some equivalences, and inequalities

The following equivalences hold for any two formulas P(x) and Q(x):

(i) $(\forall x)\ P(x) \wedge (\forall x)\ Q(x) = (\forall x)\ (P(x) \wedge Q(x))$

(ii) $(\exists x)\ P(x) \vee (\exists x)\ Q(x) = (\exists x)\ (P(x) \vee Q(x)$

But the following inequalities hold, in general:

(iii) $(\forall x)\ (P(x) \vee Q(x) \neq (\forall x)\ P(x) \vee (\forall x)\ Q(x)$
(iv) $(\exists x)\ (P(x) \wedge Q(x) \neq (\exists x)\ P(x) \wedge (\exists x)\ Q(x)$

We justify (iii) & (iv) below:
Let P(x): x is odd natural number,

Q(x): x is even natural number.

Then L.H.S of (iii) above states *for every natural number it is either odd or even,* **which is correct**. But R.H.S of (iii) states that *every natural number is odd* or *every natural number is even*, **which is not correct.**

Next, L.H.S. of (iv) states that: there is a natural number which is both even and odd, **which is not correct.** However, R.H.S. of (iv) says *there is an integer which is odd and there is an integer which is even,* correct.

Equivalences involving Negation of Quantifiers

(v) $\sim (\forall x)\ P(x) = (\exists x) \sim P(x)$
(iv) $\sim (\exists x)\ P(x) = (\forall x) \sim P(x)$

Examples: For each of the following closed formula, Prove

(i) $(\forall x)\ P(x) \wedge (\exists y) \sim P(y)$ is inconsistent.

(ii) $(\forall x)\ P(x) \rightarrow (\exists y)\ P(y)$ is valid

Solution: (i) Consider

$(\forall x)\ P(x) \wedge (\exists y) \sim P(y)$

$= (\forall x)\ P(x) \wedge \sim (\forall y)\ P(y)$ (*taking negation out*)

But we know for each bound occurrence, a variable is dummy, and can be replaced in the whole scope of the variable uniformly by another free variable. Hence,

$R = (\forall x)\ P(x) \wedge \sim (\forall x)\ P(x)$

Each conjunct of the formula is either

True of False and, hence, can be thought of as a formula of PL, in stead of formula of FOPL, Let us replace $(\forall x)\ (P(x)$ by Q , a formula of PL.

$R = Q \wedge \sim Q =$ False

Hence, the proof.

(ii) Consider

$(\forall x)\ P(x) \rightarrow (\exists y)\ P(y)$

Replacing '$\rightarrow$' we get

$= \sim (\forall x)\ P(x) \vee (\exists y)\ P(y)$

$= (\exists x) \sim P(x) \vee (\exists y)\ P(y)$

$= (\exists x) \sim P(x) \vee (\exists x)\ P(x)$ *(renaming x as y in the second disjunct)*

In other words,

$= (\exists x)\ (\sim P(x) \vee P(x))$ *(using equivalence)*

The last formula states: *there is at least one element say b, for $\sim P(b) \vee P(b)$ holds i.e., for b, either P(b) is False or P(b) is True.*

But, as P is a predicate symbol and b is a constant $\sim P(b) \vee P(b)$ must be True. Hence, the proof.

Check Your Progress - 1

Ex. 1 Let P(x) and Q(x) represent "x is a rational number" and "x is a real number," respectively. Symbolize the following sentences:

(i) Every rational number is a real number.

(ii) Some real numbers are rational numbers.

(iii) Not every real number is a rational number.

Ex. 2 Let C(x) mean "x is a used-car dealer," and H(x) mean "x is honest." Translate each of the following into English:

(i) $(\exists x)C(x)$

(ii) $(\exists x)\ H(x)$

(iii) $(\forall x)C(x) \rightarrow \sim H(x))$

(iv) $(\exists x)(C(x) \wedge H(x))$

(v) $(\exists x)(H(x) \rightarrow C(x))$.

Ex. 3 Prove the following:

(i) $P(a) \rightarrow \sim((\exists x) P(x))$ is consistent.

(ii) $(\forall x) P(x) \vee ((\exists y) \sim P(y))$ is valid.

5.5 INFERENCING & ENTAILMENT IN FOPL

In the previous unit, we discussed eight inferencing rules of Propositional Logic (PL) and further discussed applications of these rules in exhibiting validity/invalidity of arguments in **PL.** In this section, the earlier eight rules are extended to include four more rules involving quantifiers for inferencing. Each of the new rules, is called a **Quantifier Rule**. The extended set of 12 rules is then used for validating arguments in First Order Predicate Logic (**FOPL)**.

Before introducing and discussing the Quantifier rules, we briefly discuss why, at all, these rules are required. For this purpose, let us recall the argument discussed earlier, which Propositional Logic could not handle:

(i) Every man is mortal.
(ii) Raman is a man.
(iii) Raman is mortal.

The equivalent symbolic form of the argument is given by:

(i') $(\forall x)$ (Man (x) $\rightarrow$ Mortal (x)

(ii') Man (Raman)

(iii') Mortal (Raman)

If, instead of (i') we were given

(iv) Man (Raman) $\rightarrow$ Mortal (Raman) ,

(which is a formula of Propositional Logic also)

then using Modus Ponens on (ii') & (iv) in *Propositional Logic*, we would have obtained (iii') *Mortal (Raman).*

However, from (i') & (ii') we cannot derive in Propositional Logic (iii'). This suggests that there should be mechanisms for dropping and introducing quantifier appropriately, i.e., in such a manner that *validity* of arguments is not violated. Without discussing the validity-preserving characteristics, we introduce the four Quantifier rules.

(i) Universal Instantiation Rule (U.I.):

$$\frac{(\forall x)\, p(x)}{p(a)}$$

Where is an a arbitrary constant.

The rule states if ***($\forall$x) p(x)*** *is True, then we can assume* ***P(a)*** *as True for any constant* ***a*** *(where a constant* ***a*** *is like Raman). It can be easily seen that the rule associates a formula* ***P(a) of Propositional Logic*** *to a formula* ***($\forall$x) p(x) of FOPL****. The significance of the rule lies in the fact that once we obtain a formula like* ***P(a)****, then the reasoning process of Propositional Logic may be used. The rule may be used , whenever, its application seems to be appropriate.*

(ii) Universal Generalisation Rule (U.G.)

$$\frac{P(a), \text{ for all } a}{(\forall x)\, p(x)}$$

The rule says that if it is known that for all constants a, the statement P(a) is True, then we can, instead, use the formula $(\forall x)p(x)$.

The rule associates with a set of formulas ***P(a) for all*** **a of Propositional Logic**, a formula $(\forall x)p(x)$ *of FOPL.*

Before using the rule, we must ensure that P(a) is True for all a,
Otherwise it may lead to wrong conclusions.

(iii) Existential Instantiation Rule (E. I.)

$$\frac{(\exists x)\, P(x)}{P(a)} \qquad (E.I.)$$

The rule says if the Truth of $(\exists x)\; P(x)$ *is known then we can assume the Truth of P(a) for* ***some fixed*** *a . The rule, again, associates a formula P(a) of Propositional Logic to a formula* $(\forall x)p(x)$ *of FOPL.*

An inappropriate application of this rule may lead to *wrong* conclusions. The source of possible errors lies in the fact that the choice 'a' in the rule *is not arbitrary* and can not be known at the time of deducing P(a) from $(\exists x)\; P(x)$.

If during the process of deduction some other $(\exists y)\; Q(y)$ or $(\exists x)\,(\,R(x)\,)$ or even another *$(\exists x)P(x)$* is encountered, then each time a new constant say *b, c* etc. should be chosen to infer Q (b) from $(\exists y)\; Q(y)$ *or R(c)* from $(\exists x)\,(\,R(x)\,)$ *or P(d)* from $(\exists x)\; P(x)$.

(iv) Existential Generalization Rule (E.G)

$$\frac{P(a)}{(\exists x)P(x)} \qquad \text{(E.G)}$$

The rule states that if P(a), a formula of Propositional Logic is True, then the Truth of $(\exists x)\; P(x)$, a formula of FOPL , may be assumed to be True.

The Universal Generalisation (U.G) and Existential Instantiation rules should be applied with utmost care, however, other two rules may be applied, whenever, it appears to be appropriate.

Next, The purpose of the two rules, viz.,

(i) Universal Instantiation Rule (U. I.)

(iii) Existantial Rule (E. I.)

is to associate formulas of Propositional Logic (PL) to formulas of FOPL in a manner, the validity of arguments due to these associations, is not disturbed. Once, we get formulas of PL, then any of the eight rules of inference of PL may be used to validate conclusions and solve problems requiring logical reasoning for their solutions.

The purpose of the other Quantification rules viz. for generalisation, i.e.,

(ii) $\dfrac{P(a),\ \text{for all a}}{(\forall x)\ P(x)}$

(iv) $\dfrac{P(a)}{(\exists x)\,P(x)}$

is that the conclusion to be drawn in FOPL is not generally a formula of PL but a formula of FOPL. However, while making inference, we may be first associating formulas of PL with formulas of FOPL and then use inference rules of PL to conclude formulas in PL. But the conclusion to be made in the problem may correspond to a formula of FOPL. These two generalisation rules help us in associating formulas of FOPL with formulas of PL.

Example: Tell, supported with reasons, which one of the following is a correct inference and which one is not a correct inference.

(i) *To conclude* $F(a) \wedge G(a) \rightarrow H(a) \wedge I(a)$
from $(\forall x)\big(F(x) \wedge G(x)\big) \rightarrow H(x) \wedge I(x)$

using Universal Instantiation (U.I.)

The above inference or conclusion is *incorrect* in view of the fact that the scope of universal quantification is only the formula: $F(x) \wedge G(x)$ and not the whole of the formula.

The occurrences of x in $H(x) \wedge I(x)$ are free occurrences. Thus, one of the correct inferences would have been:

$$F(a) \wedge G(a) \rightarrow H(x) \wedge I(x)$$

(ii) *To conclude* $F(a) \wedge G(a) \rightarrow H(a) \wedge I(a)$ *from*
$(\forall x)\ (F(x) \wedge G(x) \rightarrow H(x) \vee I(x))$ *using U.I.*

The conclusion is correct in view of the argument given in (i) above.

(iii) *To conclude* $\sim F(a)$ *for an arbitrary a, from* $\sim (\forall x) F(x)$ *using U.I.*

The conclusion is incorrect, because actually

$$\sim (\forall x)\, F(x) = (\exists x) \sim F(x)$$

Thus, the inference is not a case of U.I., but of Existential Instantiation (E.I.)

Further, as per restrictions, we can not say for which a, $\sim F(x)$ is True. Of course, $\sim F(x)$ is true for some constant, but not necessarily for a pre-assigned constant *a*.

(iv) *to conclude* $\left((F(b) \wedge G(b) \rightarrow H(c)\right)$

from $(\exists x)\left((F(b) \wedge G(x)\right) \rightarrow H(c)$

Using E.I. is *not* correct

The reason being that the constant to be substituted for *x* cannot be assumed to be the same constant *b,* being given in advance, as an argument of *F.* However,

to conclude $\left((F(b) \wedge G(a) \rightarrow H(c)\right)$

from $(\exists x)\left((F(b) \wedge G(x)\right) \rightarrow H(c))$ is correct.

Step for using Predicate Calculus as a Language for Representing Knowledge & for Reasoning:

Step 1: Conceptualisation: First of all, all the relevant entities and the relations that exist between these entities are explicitly enumerated. Some of the implicit facts like, '*a person dead once is dead for ever*' have to be explicated.

Step 2: ***Nomenclature*** **& Translation:** Giving appropriate names to objects and relations. And then translating the given sentences given in English to formulas in FOPL. Appropriate names are essential in order to guide a reasoning system based on FOPL. It is well-established that no reasoning system is complete. In other words, a reasoning system may need help in arriving at desired conclusion.

Step 3: Finding appropriate sequence of reasoning steps, involving selection of appropriate rule and appropriate FOPL formulas to which the selected rule is to be applied, to reach the conclusion.

Applications of the 12 inferrencing rules (8 of Propositional Logic and 4 involving Quantifiers.)

Example: Symbolize the following and then construct a proof for the argument:

(i) Anyone who repairs his own car is highly skilled and saves a lot of money on repairs
(ii) Some people who repair their own cars have menial jobs. Therefore,
(iii) Some people with menial jobs are highly skilled.

Solution: Let us use the notation:

P(x) : x is a person

$S(x)$	:	x saves money on repairs
$M(x)$	:	x has a menial job
$R(x)$	:	x repairs his own car
$H(x)$	:	x is highly skilled.

Therefore, (i), (ii) and (iii) can be symbolized as:

(i) $(\forall x)\ (R(x) \rightarrow (H(x) \wedge S(x)))$
(ii) $\exists(x)\ (R(x) \wedge M(x))$
(iii) $(\exists x)\ (M(x) \wedge H(x))$ *(to be concluded)*

From (ii) using Existential Instantiation (E.I), we get, for some fixed a

(iv) $R(a) \wedge M(a)$

Then by simplification rule of Propositional Logic, we get

(v) $R(a)$

From (i), using Universal Instantiation (U.I.), we get

(vi) $R(a) \rightarrow H(a) \wedge S(a)$

Using modus ponens w.r.t. (v) and (vi) we get

(vii) $H(a) \wedge S(a)$

By specialisation of (vii) we get

(viii) $H(a)$

By specialisation of (iv) we get

(ix) $M(a)$

By conjunctions of (viii) & (ix) we get

$M(a) \wedge H(a)$

By Existential Generalisation, we get

$(\exists x)\ (M(x) \wedge H(x))$

Hence, (iii) is concluded.

Example:

(i) Some juveniles who commit minor offences are thrown into prison, and any juvenile thrown into prison is exposed to all sorts of hardened criminals.
(ii) A juvenile who is exposed to all sorts of hardened criminals will become bitter and learn more techniques for committing crimes.
(iii) Any individual who learns more techniques for committing crimes is a menace to society, if he is bitter.
(iv) Therefore, some juveniles who commit minor offences will be menaces to the society.

Example: Let us symbolize the statement in the given argument as follows:

(i) J(x) : x is juvenile.

(ii) C(x) : x commits minor offences.

(iii) P(x) : x is thrown into prison.

(iv) E(x) : x is exposed to hardened criminals.

(v) B(x) : x becomes bitter.

(vi) T(x) : x learns more techniques for committing crimes.

(vii) M(x) : x is a menace to society.

The statements of the argument may be translated as:

(i) $(\exists x)\ (J(x) \wedge C(x) \wedge P(x)) \wedge ((\forall y)\ (J(y) \rightarrow E(y))$
(ii) $(\forall x)\ (J(x) \wedge E(x) \rightarrow B(x) \wedge T(x))$
(iii) $(\forall x)\ (T(x) \wedge B(x) \rightarrow M(x))$
Therefore,

(iv) $(\exists x)\ (J(x) \wedge C(x) \wedge M(x))$

By simplification (i) becomes

(v) $(\exists x)\ (J(x) \wedge C(x) \wedge P(x))$ and
(vi) $(\forall y)\ (J(y) \rightarrow E(y))$

From (v) through Existential Instantiation, for some fixed *b*, we get

(vii) $J(b) \wedge C(b) \wedge P(b)$
Through simplification (vii) becomes

(viii) $J(b)$
(ix) $C(b)$ and
(x) $P(b)$
Using Universal Instantiation, on (vi), we get

(xi) $J(b) \rightarrow E(b)$
Using Modus Ponens in (vii) and (xi) we get

(xii) $E(b)$
Using conjunction for (viii) & (xii) we get

(xiii) $J(b) \wedge E(b)$
Using Universal Instantiation on (ii) we get

(xiv) $J(b) \wedge E(b) \rightarrow B(b) \wedge T(b)$
Using Modus Ponens for (xiii) & (xiv), we get

(xv) $T(b) \wedge B(b)$

Using Universal Instantiation for (iii) we get

(xvi) $T(b) \wedge B(b) \rightarrow M(b)$

Using Modus Ponens with (xv) and (xvi) we get

(xvii) $M(b)$

Using conjunction for (viii), (ix) and (xvii) we get

(xviii) $J(b) \wedge C(b) \wedge M(b)$

From (xviii), through Existential Generalization we get the required (iv), i.e.

$(\exists x)\ (J(x) \wedge C(x) \wedge M(x))$

Remark: It may be noted the occurrence of quantifiers is not, in general, commutative i.e.,

$(Q_1x)\ (Q_2x) \neq (Q_2x)\ (Q_1x)$

For example

$$(\forall x)\ (\exists y)\ F(x,y) \neq (\exists y)\ (\forall x)\ F(x,y) \qquad \text{(A)}$$

The occurrence of $(\exists y)$ on L.H.S depends on *x* i.e., occurrence of *y* on L.H.S is a function of *x*. However, the occurrence of $(\exists y)$ on R.H.S is independent of *x*, hence, occurrence of *y* on R.H.S is not a function of *x*.

For example, if we take F(x,y) to mean:

y and x are integers such that y>x,

then*, L.H.S of (A)* above states: *For each x there is a y such that y>x.*

The statement is true in the domain of real numbers.

On the other hand, *R.H.S of (A)* above states that: There is an integer *y* which is greater than *x*, for all *x*.

This statement is not true in the domain of real numbers.

When the logical statements are interconnected in a manner that one is consequence of other then such Logical consequences (also called entailment) are the fundamental concept in logical reasoning, which describes the relationship between statements that hold true when one statement logically follows from one or more statements.

A valid logical argument is one in which the conclusion is entailed by the premises, because the conclusion is the consequence of the premises. The philosophical analysis of logical consequence involves the questions: In what sense does a conclusion follow from its premises? and What does it mean for a conclusion to be a consequence of premises? All of philosophical logic is meant to provide accounts of the nature of logical consequence and the nature of logical truth.

Logical consequence is necessary and formal, by way of examples that explain with formal proof and models of interpretation. A sentence is said to be a logical consequence of a set of sentences, for a given language, if and only if, using only logic (i.e., without regard to any personal interpretations of the sentences) the sentence must be true if every sentence in the set is true.

5.6 CONVERSION TO CLAUSAL FORM

In order to facilitate problem solving through Propositional Logic, we discussed two normal forms, viz, the conjunctive normal form **CNF** and the disjunctive normal form **DNF**. In **FOPL**, there is a normal form called the **prenex normal form**. Further the statement in Prenex Normal Form is required to be skolomized to get the clausal form, which can be used for the purpose of Resolution.

So, first step towards the Clausal form is to begin with Prenex Normal Form (PNF), and the second step is skolomization, which will be discussed after PNF.

Prenex Normal Form (PNF): In broad sense it relates to re-alignment of the quantifiers, i.e. to bring all the quantifiers in the beginning of the expression and then replacement the existential and universal quantifiers with constants and the functions is performed for skolomization i.e. to bring the statement in the clausal form.

The use of a prenex normal form of a formula simplifies the proof procedures, to be discussed.

Definition A formula G in FOPL is said to be in a **prenex normal form** if and only if the formula G is in the form

$$(Q_1x_1)\ldots.(Q_n\ x_n)\ P$$

where each (Q_ix_i), for i = 1, ….,n, is either $(\forall x_i)$ or $(\exists x_i)$, and P is a quantifier free formula. The expression $(Q_1x_1)\ldots.(Q_n\ x_n)$ is called the **prefix** and P is called the **matrix of the formula G**.

Examples of some formulas in prenex normal form:

(i) $(\exists x)\ (\forall y)\ (R(x, y) \vee Q(y))$, $(\forall x)\ (\forall y)\ (\sim P(x, y) \rightarrow S(y))$,

(ii) $(\forall x)\ (\forall y)\ (\exists z)\ (P(x, y) \rightarrow R\ (z))$.

Next, we consider a method of transforming a given formula into a prenex normal form. For this, first we discuss equivalence of formulas in FOPL. Let us recall that two formulas E and G are **equivalent**, denoted by **E = G**, *if and only if the truth values of F and G are identical under every interpretation.* The pairs of equivalent formulas given in Table of equivalent Formulas of previous unit are still valid as these are quantifier–free formulas of FOPL. However, there are pairs of equivalent formulas of FOPL that contain quantifiers. Next, we discuss these additional pairs of equivalent formulas. We introduce some notation specific to FOPL: *the symbol G denote a formula that does not contain any free variable x*. Then we have the following pairs of equivalent formulas, where Q denotes a quantifier which is either ∀ or ∃. Next, we introduce four laws for **pairs of equivalent formulas**.

In the rest of the discussion of FOPL, P[x] is used to denote the fact that *x* is a free variable in the formula P, for example, P[x] = (∀y) P (x, y). Similarly, R [x, y] denotes that variables *x* and *y* occur as free variables **in the formula** R Some of these equivalences, we have discussed earlier.

Then, the following laws involving quantifiers hold good in FOPL

(i) $(Qx)\ P[x] \vee G = (Qx)\ (P[x] \vee G)$.

(ii) $(Qx)\ P[x] \wedge G = (Qx)\ (P[x] \wedge G)$.

In the above two formulas, Q may be either ∀ or ∃.

(iii) $\sim ((\forall x)\, P[x]) = (\exists x)\, (\sim P[x])$.

(iv) $\sim ((\exists x)\, P[x]) = (\forall x)\, (\sim P[x])$.

(v) $(\forall x)\, P[x] \wedge (\forall x)\, H[x] = (\forall x)\, (P[x] \wedge H[x])$.

(vi) $(\exists x)\, P[x] \vee (\exists x)\, H[x] = (\exists x)\, (P[x] \vee H[x])$.

That is, the universal quantifier $\forall$ and the existential quantifier $\exists$ can be distributed respectively over $\wedge$ and $\vee$.

But we must be careful about *(we have already mentioned these inequalities)*
(vii) $(\forall x)\, E[x] \vee (\forall x)\, H[x] \neq (\forall x)\, (P[x] \vee H[x])$ and
(viii) $(\exists x)\, P[x] \wedge (\exists x)\, H[x] \neq (\exists x)\, (P[x] \wedge H[x])$

Steps for Transforming an FOPL Formula into Prenex Normal Form

Step 1 Remove the connectives '$\leftrightarrow$' and '$\rightarrow$' using the equivalences

$$P \leftrightarrow G = (P \rightarrow G) \wedge (G \rightarrow P)$$

$$P \rightarrow G = \sim P \vee G$$

Step 2 Use the equivalence to remove even number of ~'s

$$\sim (\sim P) = P$$

Step 3 Apply De Morgan's laws in order to bring the negation signs immediately before atoms.

$$\sim (P \vee G) = \sim P \wedge \sim G \qquad \sim (P \wedge G) = \sim P \vee \sim G$$

and the quantification laws

$$\sim ((\forall x)\, P[x]) = (\exists x)\, (\sim P[x]) \qquad \sim ((\exists x)\, P[x]) = (\forall x)\, (\sim F[x])$$

Step 4 rename bound variables **if necessary**

Step 5 Bring quantifiers to the left before any predicate symbol appears in the formula. This is achieved by using (i) to (vi) discussed above.

We have already discussed that, if all occurrences of a bound variable are replaced uniformly throughout by another variable not occurring in the formula, then the equivalence is preserved. Also, we mentioned under (vii) that $\forall$ does not distribute over $\wedge$ and under (viii) that $\exists$ does not distribute over $\vee$. In such cases, in order to bring quantifiers to the left of the rest of the formula, we may have to first rename one of bound variables, say x, may be renamed as z, which does not occur either as free or bound in the other component formulas. And then we may use the following equivalences.

$$(Q1\ x)\, P[x] \vee (Q2\ x)\, H[x] = (Q1\ x)\, (Q2\ z)\, (P[x] \vee H[z])$$

$$(Q3\ x)\, P[x] \wedge (Q4\ x)\, H[x] = (Q3\ x)\, (Q4\ z)\, (P[x] \wedge H[z])$$

Example: Transform the following formulas into prenex normal forms:

(i) $(\forall x)\,(Q(x) \rightarrow (\exists x)\, R\,(x, y))$
(ii) $(\exists x)\,(\sim (\exists y)\, Q(x, y) \rightarrow ((\exists z)\, R(z) \rightarrow S\,(x)))$
(iii) $(\forall x)\,(\forall y)\,((\exists z)\, Q(z, y, z) \wedge ((\exists u)\, R\,(x, u) \rightarrow (\exists v)\, R\,(y, v)))$.

Part (i)

Step 1: *By removing '→', we get*

$$(\forall x)\,(\sim Q\,(x) \vee (\exists x)\, R\,(x, y))$$

Step 2: *By renaming x as z in (∃x) R (x, y) the formula becomes*

$$(\forall x)\,(\sim Q\,(x) \vee (\exists z)\, R\,(z, y))$$

Step 3: *As ~ Q(x) does not involve z, we get*

$$(\forall x)\,(\exists z)\,(\sim Q\,(x) \vee R\,(z, y))$$

Part (ii)

$(\exists x)\,(\sim (\exists y)\, Q\,(x, y) \rightarrow ((\exists z)\, R\,(z) \rightarrow S\,(x)))$

Step 1: Removing outer '→' we get

$$(\exists x)\,(\sim (\sim ((\exists y)\, Q\,(x, y))) \vee ((\exists z)\, R\,(z) \rightarrow S\,(x)))$$

Step 2: Removing inner '→' , and simplifying ~ (~ ( )) we get

$$(\exists x)\,((\exists y)\, Q\,(x, y) \vee (\sim ((\exists z)\, R(z)) \vee S\,(x)))$$

Step 3: Taking '~' inner most, we get

$$(\exists x)\,(\exists y)\, Q\,(x, y) \vee ((\forall z) \sim R(z) \vee S(x)))$$

As first component formula Q (x, y) does not involve z and S(x) does not involve both y and z and ~ R(z) does not involve y. Therefore, we may take out (∃ y) and (∀z) so that, we get

$(\exists x)\,(\exists y)\,(\forall z)\,(Q\,(x, y) \vee (\sim R(z) \vee S\,(x)\,)$, which is the required formula in prenex normal form.

Part (iii)

$(\forall x)\,(\forall y)\,((\exists z)\, Q\,(x, y, z) \wedge ((\exists u)\, R\,(x, u) \rightarrow (\exists v)\, R\,(y\ v)))$

Step 1: Removing '→', we get

$$(\forall x)\,(\forall y)\,((\exists z)\, Q\,(x, y, z) \wedge (\sim ((\exists u)\, R\,(x, u)) \vee (\exists v)\, R\,(y, v)))$$

Step 2: Taking '~' inner most, we get

$$(\forall x)\,(\forall y)\,((\exists z)\, Q\,(x, y, z) \wedge ((\forall u) \sim R\,(x, u) \vee (\exists v)\, R\,(y, v)))$$

Step 3: As variables z, u & v do not occur in the rest of the formula except the formula which is in its scope, therefore, we can take all quantifiers outside, preserving the order of their occurrences, Thus we get

$(\forall x)\ (\forall y)\ (\exists z)\ (\forall u)\ (\exists v)\ (Q\ (x, y, z) \wedge (\sim R\ (x, u) \vee R\ (y, v)))$

Skolomization : A further refinement of Prenex Normal Form (PNF) called (Skolem) Standard Form, is the basis of problem solving through Resolution Method. The Resolution Method will be discussed next.

The **Standard Form of a formula of FOPL is** obtained through the following three steps:

(1) The given formula should be converted to Prenex Normal Form (PNF), and then

(2) Convert the Matrix of the PNF, i.e, quantifier-free part of the PNF into conjunctive normal form
(3) Skolomization: Eliminate the existential quantifiers using skolem constants and functions

Before illustrating the process of conversion of a formula of FOPL to Standard Normal Form, through examples, we discuss briefly skolem functions.

Skolem Function
We in general, mentioned earlier that $(\exists x)\ (\forall y)\ P(x,y) \neq\ (\forall y)\ (\exists x)\ P(x,y)$…….(1)

For example, if P(x,y) stands for the relation 'x>y' in the set of integers, then the L.H.S. of the inequality (i) above states: *some (fixed) integer (x) is greater than all integers (y)*. This statement is False.

On the other hand, R.H.S. of the inequality (1) states: *for each integer y, there is an integer x so that x>y*. This statement is True.

The difference in meaning of the two sides of the inequality arises because of the fact that on L.H.S. *x* in $(\exists x)$ is independent of *y* in $(\forall y)$ **whereas** on R.H.S *x* of dependent on *y*. In other words, *x* on L.H.S. of the inequality can be replaced by some constant say 'c' whereas on the right hand side *x* is some function, say, f(y) of *y*.

Therefore, the two parts of the inequality (i) above may be written as

LH.S. of (1) $= (\exists x)\ (\forall y)\ P\ (x,y) = (\forall y)\ P(c,y)$,

Dropping x because there is no x appearing in $(\forall y)\ P(c,y)$

R.H.S. of (1) $= (\forall y)\ (\exists x)\ P(f(y),y) = (\forall y)\ P(f(y), y)$

The above argument, in essence, explains what is meant by each of the terms viz. *skolem constant, skolem function and skolomisation.*

The constants and functions which replace existential quantifiers are respectively called **skolem constants and skolem functions.** The process of replacing all existential variables by skolem constants and variables is called **skolemisation**.

A form of a formula which is obtained after applying the steps for

(i) reduction to PNF and then to

(ii) CNF and then

(iii) applying skolomization is called **Skolem Standard Form** or just **Standard Form.**

We explain through examples, the skolomisation process after PNF and CNF have already been obtained.

Example: Skolomize the following:

(i) $(\exists x_1)\ (\exists x_2)\ (\forall y_1)\ (\forall y_2)(\exists x_3)(\forall y_3)\ P(x_1, x_2, x_3, y_1, y_2, y_3)$

(ii) $(\exists x_1)(\forall y_1)(\exists x_2)(\forall y_2)\ (\exists x_3)P(x_1, x_2, x_3, y_1, y_2\ )\wedge(\exists x_1)(\forall y_3)(\ \exists x_2)\ (\forall y_4)Q(x_1, x_2,$

$y_3, y_4)$

Solution (i) As existential quantifiers x_1 and x_2 precede all universal quantifiers, therefore, x_1 and x_2 are to be replaced by *constants*, but by distinct constants, say by 'c' and 'd' respectively. As existential variable x_3 is preceded by universal quantifiers y_1 and y_2, therefore, x_3 is replaced by some function $f(y_1, y_2)$ of the variables y_1 and y_2. After making these substitutions and dropping universal and existential variables, we get the skolemized form of the given formula as

$(\forall y_1)\ (\forall y_2)\ (\forall y_3)\ (c, d, f\ (y_1, y_2), y_1, y_2, y_3)$.

Solution (ii) As a first step we must bring all the quantifications in the beginning of the formula through Prenex Normal Form reduction. Also,

$$(\exists x)\ldots P(x,\ldots) \wedge (\exists x)\ldots.\ Q\ (x,\ldots.) \neq (\exists x)\ (\ldots.P(x) \wedge \ldots Q(x,\ldots.),$$

therefore, we rename the second occurrences of quantifiers $(\forall x_1)$ and $(\forall x_2)$ by renaming these as x_5 and x_6. Hence, after renaming and pulling out all the quantifications to the left, we get

$$(\exists x_1)\ (\forall y_1)\ (\exists x_2)\ (\forall y_2)\ (\exists x_3)\ (\exists x_5)\ (\forall y_3)\ (\exists x_6)\ (\forall y_4)$$

$$(P(x_1, x_2, x_3, y_1, y_2) \wedge Q\ (x_5, x_6, y_3, y_4)$$

Then the existential variable x_1 is independent of all the universal quantifiers. Hence, x_1 may be replaced by a constant say, 'c'. Next x_2 is preceded by the universal quantifier y_1 hence, x_2 may be replaced by $f\ (y_1)$. The existential quantifier x_3 is preceded by the universal quantifiers y_1 and y_2. Hence x_3 may be replaced by g

(y_1, y_2). The existential quantifier x_5 is preceded by again universal quantifier y_1 and y_2. In other words, x_5 is also a function of y_1 and y_2. But, we have to use a different function symbol say h and replace x_5 by $h\ (y_1, y_2)$. Similarly x_6 may be replaced by

$j\ (y_1, y_2, y_3)$.

Thus, (Skolem) Standard Form becomes
$(\forall y_1)\ (\forall y_2)\ (\forall y_3)\ (P\ (c, f(y_1), g(y_1, y_2), y_1, y_2) \wedge Q\ (h\ (y_1, y_2), j\ (y_1, y_2, y_3)))$.

Check Your Progress -2

Ex: 4 (i) Transform the formula $(\forall x)\ P(x) \rightarrow (\exists x)\ Q(x)$ into prenex normal form.

(ii) Obtain a prenex normal form for the formula

$(\forall x)\ (\forall y)\ ((\exists z)\ (P(x, y) \wedge P(y, z)) \rightarrow (\exists u)\ Q\ (x, y, u))$

Ex 5. Obtain a (skolem) standard form for each of the following formula:

(i) $(\exists x)\ (\forall y)\ (\forall v)\ (\exists z)\ (\forall w)\ (\exists u)\ P\ (x, y, z, u, v, w)$
(ii) $(\forall x)\ (\exists y)\ (\exists z)\ ((P\ (x, y) \vee \sim Q\ (x, z)) \rightarrow R\ (x, y, z))$

5.7 RESOLUTION & UNIFICATION

In the beginning of the previous section, we mentioned that resolution method for FOPL requires discussion of a number of complex new concepts. Also, , we discussed (Skolem) Standard Form and also discussed how to obtain Standard Form for a given formula of FOPL. In this section, along with Resolution we will introduce two new, and again complex, concepts, viz., *substitution and unification.*

The complexity of the resolution method for FOPL mainly results from the fact that a clause in FOPL is generally of the form : $P(x) \vee Q(f(x), x, y) \vee \ldots$, in which the variables x, y, z, may assume any one of the values of their domain.

Thus, the atomic formula $(\forall x)\ P(x)$, *which after dropping of universal quantifier, is written as just P(x)* stands for $P(a_1) \wedge P(a_2) \ldots \wedge P(a_n)$ where the set $\{a_1\ a_2 \ldots, a_n\}$ is assumed here to be domain (x).

Similarly, $(\exists x)\ P(x)$ *stands for* $(P(a_1) \vee P(a_2) \vee \ldots \vee P(a_n))$

However, in order to resolve two clauses – one containing say P(x) and the other containing ~ P(y) where *x* and *y* are universal quantifiers, possibly having some restrictions, we have to know which values of x and y satisfy both the clauses. For this purpose we need the concepts of **substitution** and **unification** as defined and discussed in the rest of the section.

Instead of giving formal definitions of substitution, unification, unifier, most general unifier and resolvent, resolution of clauses in FOPL, we illustrate the concepts through examples and minimal definitions, if required

Example: Let us consider our old problem:

To conclude

(i) Raman is mortal

From the following two statements:

(ii) Every man is mortal and

(iii) Raman is a man

Using the notations

MAN (x) : x is a man

MORTAL (x) : x is mortal,

the problem can be formulated in symbolic logic as: Conclude

MORTAL (Raman)

from

(ii) $((\forall x)\ (MAN(x) \rightarrow MORTAL\ (x))$

(iii) MAN (Raman).

As resolution is a refutation method, assume

(i) ~ MORTAL (Raman)

After Skelomization and dropping ($\forall$x), (ii) in standard form becomes

(i) ~ MAN (x) $\vee$ MORTAL (x)
(ii) MAN (Raman)

In the above *x* varies over the set of human beings including Raman. Hence, one special instance of (iv) becomes

(vi) ~ MAN (Raman) $\vee$ MORTAL (Raman)

At the stage, we may observe that

(a) *MAN(Raman)* and *MORTAL(Raman)* do not contain any variables, and, hence, their truth or falsity can be determined directly. Hence, each of like a formula of PL. In term of formula which does not contain any variable is called **ground term** or **ground formula**.

(b) Treating *MAN (Raman)* as formula of PL and using resolution method on (v) and (vi), we conclude

(vii) MORTAL (Raman),

Resolving (i) and (vii), we get **False**. Hence, the solution.

Unification: In the process of solution of the problem discussed above, we tried to make the two expression *MAN(x)* and *MAN(Raman)* identical. Attempt to make identical two or more expressions is called *unification.*

In order to unify *MAN (x)* and *MAN (Raman)* identical, we found that because one of the possible values of *x* is *Raman* also. And, hence, we replaced *x* by one of its possible values : *Raman*.

This replacement of a variable like *x*, by a term (*which may be another variable also*)

which is one of the possible values of *x*, is called **substitution**. The substitution, in this case is denoted formally as {Raman/x}

Substitution, in general, **notationally** is of the form $\{t_1 / x_1, t_2 / x_2 \ldots t_m / x_m\}$ where $x_1, x_2 \ldots, x_m$ are variables and $t_2, t_2 \ldots t_m$ are terms and t_i replaces the variable x_i in some expression.

Example: (i) Assume Lord Krishna is loved by everyone who loves someone (ii) Also assume that no one loves nobody. Deduce Lord Krishna is loved by everyone.

Solution: Let us use the symbols

Love (x, y): *x loves y (or y is loved by x)*

LK : *Lord Krishna*

Then the given problem is formalized as :

(i) ($\forall$x) (($\exists$y) Love (x, y)$\rightarrow$Love (x, LK))

(ii) $\sim (\exists x) ((\forall y) \sim \text{Love}(x, y))$

To show : $(\forall x) (\text{Love}(x, LK))$

As resolution is a refutation method, assume negation of the last statement as an axiom.

(iii) $\sim (\forall x) \text{ Love}(x, LK)$

The formula in (i) above is reduced in standard form as follows:

$(\forall x) (\sim (\exists y) \text{ Love}(x, y) \vee \text{Love}(x, LK))$

$= (\forall x) ((\forall y) \sim \text{Love}(x, y) \vee \text{Love}(x, LK))$

$= (\forall x) (\forall y) (\sim \text{Love}(x, y) \vee \text{Love L}(x, LK))$

($\because$ $(\forall y)$ does not occurs in Love (x, LK))

After dropping universal quantifications, we get

(iv) $\sim \text{Love}(x, y) \vee \text{Love}(x, LK)$

Formula (ii) can be reduced to standard form as follows:

(ii) $= (\forall x) (\exists y) \text{ Love}(x, y)$

y is replaced through skolomization by f(x)

so that we get

$(\forall x) \text{ Love}(x, f(x))$

Dropping the universal quantification

(v) Love (x, f(x))

The formula in (iii) can be brought in standard form as follows:

(iii) $= (\exists x) (\sim \text{Love}(x, LK))$

As existential quantifier *x* is not preceded by any universal quantification, therefore, *x* may be substituted by a constant *a* , i.e., we use the substitution $\{a/x\}$ in (iii) to get the standard form:

(vi) $\sim$ Love (a, LK).

Thus, to solve the problem, we have the following standard form formulas for resolution:

(iv) $\sim \text{Love}(x, y) \vee \text{Love}(x, LK)$

(v) Love (x, f(x))

(vi) $\sim$ Love (a, LK).

Two possibilities of resolution exist for two pairs of formulas viz.

one possibility: resolving (v) and (vi).

second possibility : resolving (iv) and (vi).

The possibilities exist because for each possibility pair, the predicate *Love* occurs in complemented form in the respective pair.

Next we attempt to resolve (v) and (vi)

For this purpose we attempt to make the two formulas *Love(x, f(x))* and *Love (a, LK)* identical, through unification involving substitutions. We start from the left, matching the two formulas, term by term. First place where matching may fail is when 'x' occurs in one formula and 'a' occurs in the other formula. **As, one of these happens to be a variable**, hence, the substitution {a/x} can be used to unify the portions so far.

Next, possible disagreement through term-by-term matching is obtained when we get the two disagreeing terms from two formulas as f(x) and LK. **As none of f(x) and LK is a variable** *(note f(x) involves a variable but is itself not a variable),* hence, no unification and, hence, no resolution of (v) and (vi) is possible.

Next, we attempt unification of **(vi) Love (a, LK)** *with* **Love (x, LK)** of (iv).

Then first term-by-term possible disagreement occurs when the corresponding terms are 'a' and 'x' respectively. As one of these is a variable, hence, the substitution{a/x} unifies the parts of the formulas so far. Next, the two occurrences of LK, one each in the two formulas, match. Hence, the whole of each of the two formulas can be unified through the substitution {a/x}. Though the unification has been *attempted* in corresponding smaller parts, substitution has to be carried **in the whole of the formula**, in this case in whole of (iv). Thus, after substitution, (iv) becomes
(viii) ~ Love (a, y) $\vee$ Love (a, L K)

resolving (viii) with (vi) we get

(ix) ~ Love (a, y)

In order to resolve (v) and (ix), we attempt to unify **Love (*x*, f(x))** of (v) with

Love (a, y) of (ix).

The term-by-term matching leads to possible disagreement of *a* of (ix) with *x* of (v).

As, one of these is a variable, hence, the substitution {a/x} will unify the portions considered so far.

Next, possible disagreement may occur with f (x) of (v) and y of (ix). As one of these are a variable viz. y, therefore, we can unify the two terms through the substitution {f(x)/y}. Thus, the complete substitution *{a/x, f (x)/y}* is required to match the formulas. Making the substitutions, we get (v) becomes Love (a, f(x)) and (ix) becomes ~ Love (a, f (x))

Resolving these formulas we get **False.** Hence, the proof.

Check you Progress - 3

Ex. 6: Unify, if possible, the following three formulas:

(i) Q (u, f (y, z)),
(ii) Q (u, a)
(iii) Q (u, g (h (k (u))))

Ex. 7: Determine whether the following formulas are unifiable or not:

(i) Q (f (a), g(x))
(ii) Q (x, y)

Example: Find resolvents, if possible for the following pairs of clauses:

(i) ~ Q (x, z, x) ∨ Q (w, z, w) and
(ii) Q (w, h (v, v), w)

Solution: As two literals with predicate Q occur and are mutually negated in (i) and (ii),therefore, there is possibility of resolution of ~ *Q (x, z, x)* from (i) with *Q (w, h (v, v), w)* of (ii). We attempt to unify Q (x, z, x) and Q (w, h (v, v), w), if possible, by finding an appropriate substitution. First terms x and w of the two are variables, hence, unifiable with either of the substitutions {x/w} or {w/x}. Let us take {w/x}.

Next pair of terms from the two formulas, viz, z and h(v, v) are also unifiable, because, one of the terms is a variable, and the required substitution for unification is { h (v, v)/z}.

Next pair of terms at corresponding positions is again {w, x} for which, we have determined the substitution {w/x}. Thus, the substitution {w/x, h(v, v)/z} unfies the two formulas. Using the substitutions, (i) and (ii) become resp. as

(iii) ~ Q (w, h (v, v), w) ∨ Q (w, h (v, v), w)
(iv) Q (w, h (v, v), w)

Resolving, we get

Q (w, h (v, v), w),

which is the required resolvent.

5.8 SUMMARY

In this unit, initially, we discuss how PL is inadequate to solve even simple problems, requires some extension of PL or some other formal inferencing system so as to compensate for the inadequacy. First Order Predicate Logic (FOPL), is such an extension of PL that is discussed in the unit.
Next, syntax of proper structure of a formula of FOPL is discussed. In this respect, a number of new concepts including those of quantifier, variable, constant, term, free and bound occurrences of variables; closed and open wff, consistency/validity of wffs etc. are introduced.

Next, two normal forms viz. Prenex Normal Form (PNF) and Skolem Standard Normal Form are introduced. Finally, tools and techniques developed in the unit, are used to solve problems involving logical reasoning.

5.9 SOLUTIONS/ANSWERS

Check Your Progress - 1

Ex. 1 *(i)* $(\forall x)\ (P(x) \rightarrow Q(x))$

(ii) $(\exists x)\ (P(x) \wedge Q(x))$

(iii) ~ (∀x) (Q (x) → P(x))

Ex. 2

(i) There is (at least) one (person) who is a used-car dealer.

(ii) There is (at least) one (person) who is honest.

(iii) All used-car dealers are dishonest.

(iv) (At least) one used-car dealer is honest.

(v) There is at least one thing in the universe, (for which it can be said that) if that something is Honest then that something is a used-car dealer

Note: *the above translation is not the same as: Some no gap one honest, is a used-car dealer.*

Ex 3: *(i) After removal of '→' we get the given formula*

$$= \sim P(a) \vee \sim ((\exists x)\, P(x))$$

$$= \sim P(a) \vee (\forall x)\, (\sim P(x))$$

Now P(a) is an atom in PL which may assume any value T or F. On taking P(a) as F the given formula becomes T, hence, consistent.

(ii) The formula can be written

$(\forall x)\, P(x) \vee \sim (\forall x)\, (P(x))$, by taking negation outside the second disjunct and then renaming.

The $(\forall x)\, P(x)$ being closed is either T or F and hence can be treated as formula of PL.
Let $\forall x\, P(x)$ be denoted by Q. Then the given formula may be denoted by $Q \vee \sim Q$ = True (always)
Therefore, formula is valid.

<u>Check Your Progress - 2</u>

Ex: 4 (i) $(\forall x)\, P(x) \rightarrow (\exists x)\, Q(x) = \sim ((\forall x)\, P(x)) \vee (\exists x)\, Q(x)$ *(by removing the connective→)*

$$= (\exists x)\, (\sim P(x)) \vee (\exists x)\, Q(x) \text{ (by taking `$\sim$' inside)}$$
$$= (\exists x)\, (\sim P(x) \vee Q(x)) \text{ (By taking distributivity of $\exists x$ over $\vee$)}$$

Therefore, a prenex normal form of $(\forall x)\, P(x) \rightarrow (\exists x)\, Q(x)$ is $(\exists x)\, (\sim P(x) \vee Q(x))$.

(ii) $(\forall x)\, (\forall y)\, ((\exists z)\, (P(x, y) \wedge P(y, z)) \rightarrow (\exists u)\, Q\, (x, y, u))$ *(removing the connective→)*

$$= (\forall x)\, (\forall y)\, (\sim ((\exists z)\, (P(x, z) \wedge P(y, z))) \vee (\exists u)\, Q\, (x, y, u)) \quad \text{(using De Morgan's Laws)}$$

$$= (\forall x)\, (\forall y)\, ((\forall z)\, (\sim P(x, z) \vee \sim P(y, z)) \vee (\exists u)\, Q(x, y, u))$$

$$= (\forall x)\, (\forall y)\, (\forall z)\, (\sim P(x, z)$$

$\vee \sim P(y, z) \vee Q(x, y, u)$ *(as z and u do not occur in the rest of the formula except their respective scopes)*

Therefore, we obtain the last formula as a prenex normal form of the first formula.

Ex 5 (i) *In the given formula ($\exists x$) is not preceded by any universal quantification. Therefore, we replace the variable x by a (skolem) constant c in the formula and drop ($\exists x$).*
Next, the existential quantifier ($\exists z$) is preceded by two universal quantifiers viz., v and y. we replace the variable z in the formula, by some function, say, f (v, y) and drop ($\exists z$). Finally, existential variable ($\exists u$) is preceded by three universal quantifiers, viz., ($\forall y$), ($\forall y$) and ($\forall w$). Thus, we replace in the formula the variable u by, some function g(y, v, w) and drop the quantifier ($\exists u$). Finally, we obtain the standard form for the given formula as
$(\forall y)\ (\forall v)\ (\forall w)\ P(x, y, z, u, v, w)$

(ii) *First of all, we reduce the matrix to CNF.*

$= (P(x, y) \vee \sim Q(x, z)) \rightarrow R(x, y, z)$

$= (\sim P(x, y) \wedge Q(x, z)) \vee R(x, y, z)$

$= (\sim P(x, y) \vee R(x, y, z)) \wedge (Q(x, z) \vee R(x, y, z))$

Next, in the formula, there are two existential quantifiers, viz., ($\exists y$) and ($\exists z$). Each of these is preceded by the only universal quantifier, viz. ($\forall x$).

Thus, each variable y and z is replaced by a function of x. But the two functions of x for y and z must be different functions. Let us assume, variable, y is replaced in the formula by f(x) and the variable z is replaced by g(x). Thus the initially given formula, after dropping of existential quantifiers is in the standard form:
$(\forall x)\ ((\sim P(x, y) \vee R(x, y, z)) \wedge (Q(x, z) \vee R(x, y, z)))$

Check Your Progress - 3

Ex 6 : Refer to section 5.7

Ex 7 : Refer to section 5.7

5.10 FURTHER READINGS

1. Ela Kumar, " Artificial Intelligence", IK International Publications
2. E. Rich and K. Knight, "Artificial intelligence", Tata Mc Graw Hill Publications
3. N.J. Nilsson, "Principles of AI", Narosa Publ. House Publications
4. John J. Craig, "Introduction to Robotics", Addison Wesley publication
5. D.W. Patterson, "Introduction to AI and Expert Systems" Pearson publication
6. McKay, Thomas J., *Modern Formal Logic* (Macmillan Publishing Company, 1989).
7. Gensler, Harry J. *Symbolic Logic: Classical and Advanced Systems* (Prentice Hall of India, 1990).
8. Klenk, Virginia *Understanding Symbolic Logic* (Prentice Hall of India 1983)
9. Copi Irving M. & Cohen Carl, *Introduction Logic IX edition*, (Prentice Hall of India, 2001).

UNIT 6 RULE BASED SYSTEMS AND OTHER FORMALISM

Structure **Page Nos.**

6.0 INTRODUCTION

Computer Science is the study of how to create models that can be represented in and executed by some computing equipment. In this respect, the task for a computer scientist is to create, in addition to a model of the problem domain, a model of an expert of the domain as problem solver who is highly skilled in solving problems from the domain under consideration, and the concerned field relates to the field of Expert Systems.

First of all we must understand that an expert system is nothing but a computer program or a set of computer programs which contains the knowledge and some inference capability of an expert, most generally a human expert, in a particular domain. An expert system is supposed to contain the capability to lead to some conclusion, based on the inputs provided, the system already contains some pre-existing information; which is processed to infer some conclusion. The expert system belongs to the branch of Computer Science called Artificial Intelligence.

Taking into consideration all the points, discussed above, one of the many possible definitions of an Expert System is *: "An Expert System is a computer program that possesses or represents knowledge in a particular domain, has the capability of processing/ manipulating or reasoning with this knowledge with a view to solving a problem, giving some achieving or to achieve some specific goal."*

Whereas, *the Artificial Intelligence* programs written to achieve expert-level competence in solving problems of different domains are more called **knowledge based systems**. **A knowledge-based system** is any system which performs a job or task by applying rules of thumb to a symbolic representation of knowledge, instead of employing mostly algorithmic or statistical methods. Often the term *expert systems* is reserved for programs whose knowledge base contains the knowledge used by human experts, in contrast to knowledge gathered from textbooks or non-experts. **But more often than not, the two terms, expert systems and knowledge-based systems are taken as synonyms.** Together they represent the most widespread type of *AI* application.

One of the underlying assumptions in Artificial Intelligence is that **intelligent behaviour can be achieved through the manipulation of symbol structures** (representing bits of knowledge). One of the main issues in *AI* is to find appropriate representation of problem elements and available actions as symbol structures so that the representation can be used to intelligently solve problems. In *AI*, **an important criteria about knowledge representation schemes or languages is that they should support *inference***. For intelligent action, the inferencing capability is essential in view of the fact that we can't represent explicitly everything that the system might ever need to know–**some things have to be left implicit, to be inferred/deduced by the system** as and when needed in problem solving.

In general, a good knowledge representation scheme should have the following features:

- It should allow us to express the knowledge we wish to represent in the language. For example, the mathematical statement: *Every symmetric and transitive relation on a domain, need not be reflexive* is not expressible in First Order Logic.
- It should allow new knowledge to be inferred from a basic set of facts, as discussed above.
- It should have well-defined *syntax* and *semantics*.

Building a expert system is known as **knowledge engineering** and its practitioners are called **knowledge engineers**. It is the job of the knowledge engineer to ensure to make sure that the computer has all the knowledge needed to solve a problem. **The knowledge engineer must choose** one or more forms in which to represent the required knowledge i.e., s/he must choose **one or more knowledge representation schemes**.

A number of knowledge representing schemes like predicate logic, semantic nets, frames, scripts and rule based systems, exists; and we will discuss them in this unit. Some popular knowledge representation schemes are:

- *First order logic,*
- *Semantic networks,*
- *Frames,*
- *Scripts and,*
- *Rule-based systems.*

As predicate logic have been discussed in previous blocks so we will discuss the remaining knowledge representation schemes here in this unit.

6.1 OBJECTIVES

After going through this unit, you should be able to:

- Understand the basics of expert system
- Understand the basics of Knowledge based systems
- discuss the various knowledge representation scheme like rule based systems, semantic nets, frames, and scripts

6.2 RULE BASED SYSTEMS

We know that **Planning** is the process that exploits the structure of the problem under consideration for designing a sequence of actions in order to solve the problem under consideration.

In order to plan a solution to the problem, one should have the knowledge of the nature and the structure of the problem domain, under consideration. For the purpose of planning, the problem environments are divided into two categories, viz., classical planning environments and non-classical planning environments. The **classical** planning environments/domains are fully observable, deterministic, finite, static and discrete. On the other hand, **non-classical** planning environments may be only partially observable and/or stochastic. In this unit, we discuss planning only for classical environments.

Let's begin with the Rule Based Systems :

Rather than representing knowledge in a declarative and somewhat static way (as a set of statements, each of which is true), rule-based systems represent knowledge in terms of a set of rules each of which specifies the conclusion that could be reached or derived under given conditions or in different situations. A rule-based system consists of

(i) Rule base, which is a set of IF-THEN *rules*,
(ii) A bunch of *facts*, and
(iii) Some *interpreter* of the facts and rules which is a mechanism which decides which rule to apply based on the set of available facts. The interpreter also initiates the action suggested by the rule selected for application.

A Rule-base may be of the form:

R_1: If A is an animal and A barks, than A is a dog
F1: Rocky is an animal
F2: Rocky Barks

The rule-interpreter, after scanning the above rule-base may conclude: Rocky is a dog.
After this interpretation, the rule-base becomes

R_1: If A is an animal and A barks, then A is a dog
F1: Rocky is an animal
F2: Rocky Barks
F3: Rocky is a dog.

There are two broad kinds of rule-based systems:

- ***Forward chaining*** **systems**,
- ***Backward chaining*** **systems**.

In a **forward** chaining system we start with the initial facts, and keep using the rules to draw new intermediate conclusions (or take certain actions) given those facts. The process terminates when the final conclusion is established. In a **backward** chaining system, we start with some goal statements, which are intended to be established and keep looking for rules that would allow us to conclude, setting new sub-goals in the process of reaching the ultimate goal. In the next round, the subgoals become the new goals to be established. The process terminates when in this process all the subgoals are given fact. Forward chaining systems are primarily **data-driven**, while backward chaining systems are **goal-driven**. We will discuss each in detail.
Next, we discuss in detail some of the issues involved in a rule-based system.

Advantages of Rule-base

A basic principle of rule-based system is that each rule is an independent piece of knowledge. In an IF-THEN rule, the IF-part contains all the conditions for the application of the rule under consideration. THEN-part tells the action to be taken by the interpreter. The interpreter need not search any where else except within the rule itself for the conditions required for application of the rule.

Another important consequence of the above-mentioned characteristic of a rule-based system is that no rule can call upon any other and hence rules are ignorant and hence independent, of each other. This gives a highly modular structure to the rule-based systems. Because of the highly modular structure of the rule-base, the rule-based system addition, deletion and modification of a rule can be done without any danger side effects.

Disadvantages

The main problem with the rule-based systems is that when the rule-base grows and becomes very large, then checking (i) whether a new rule intended to be added is redundant, i.e., it is already covered by some of the earlier rules. Still worse, as the rule- base grows, checking the consistency of the rule-base also becomes quite difficult. By consistency, we mean there may be two rules having similar conditions, the actions by the two rules **conflict** with each other.

Let us first define working memory, before we study forward and backward chaining systems.

Working Memory: A working is a representation, in which

- lexically, there are application –specific symbols.
- structurally, assertions are list of application-specific symbols.
- semantically, assertions denote facts.
- assertions can be added or removed from working memory.

Rule based systems usually work in domains where conclusions are rarely certain, even when we are careful enough to try and include everything we can think of in the antecedent or condition parts of rules.

Sources of Uncertainty

Two important sources of uncertainty in rule based systems are:

✓ The theory of the domain may be vague or incomplete so the methods to generate exact or accurate knowledge are not known.
✓ Case data may be imprecise or unreliable and evidence may be missing or in conflict.

So even though methods to generate exact knowledge are known but they are impractical due to lack or data, imprecision or data or problems related to data collection.

So ***rule based deduction system developers often build some sort of certainty or probability computing procedure on and above the normal condition-action format of rules.*** Certainty computing procedures attach a **probability between 0 and 1** with each assertion or fact. Each probability reflects how certain an assertion is, whereas certainty factor of 0 indicates that the assertion is definitely false and certainty factor of 1 indicates that the assertion is definitely true.

Example 1: In the example discussed above the assertion (ram at-home) may have a certainty factor, say 0.7 attached to it.

Example 2: In MYCIN a rule based expert system (which we will discuss later), a rule in which statements which link evidence to hypotheses are expressed as decision criteria, may look like :

IF patient has symptoms s1,s2,s3 and s4
AND certain background conditions t1,t2 and t3 hold
THEN the patient has disease d6 with certainty 0.75

For detailed discussion on certainty factors, the reader may refer to probability theory, fuzzy sets, possibility theory, Dempster-Shafter Theory etc.

6.2.1 Forward Chaining Systems

In a forward chaining system the facts in the system are represented in a ***working memory*** which is continually updated, so on the basis of a rule which is currently being applied, the number of facts may either increase or decrease. Rules in the system represent possible actions to be taken when specified conditions hold on items in the working memory–they are sometimes called **condition-action or antecedent-consequent rules**. The conditions are usually ***patterns*** that must *match* items in the working memory, while the actions usually involve ***adding or deleting*** items from the working memory. So **we can say that in forward chaining proceeds forward, beginning with facts, chaining through rules, and finally establishing the goal**. Forward chaining systems usually represent rules in standard implicational form, with an antecedent or condition part consisting of positive literals, and a consequent or conclusion part consisting of a positive literal.

The interpreter controls the application of the rules, given the working memory, thus controlling the system's activity. It is based on a cycle of activity sometimes known as a ***recognize-act*** cycle. The system first checks to find all the rules whose condition parts are satisfied i.e., the those rules which are applicable, given the current state of working memory (**A rule is applicable if** each of the literals in its antecedent i.e., the condition part can be unified with a corresponding fact using consistent substitutions. This restricted form of unification is called pattern matching). It then selects one and performs the actions in the action part of the rule which may involve addition or deleting of facts. The actions will result in a new i.e., updated working memory, and the cycle starts again (**When more than one rule is applicable**, then some sort of external **conflict resolution scheme** is used to decide which rule will be applied. But when there are a large numbers of rules and facts then the number of unifications that must be tried becomes prohibitive or difficult). This cycle will be repeated until either there is no rule which fires, or the required goal is reached.

Rule-based systems vary greatly in their details and syntax, let us take the following example in which we use forward chaining :

Example

Let us assume that the working memory initially contains the following facts :

(day monday)
(at-home ram)
(does-not-like ram)

Let, the existing set of rules are:

R1 : IF (day monday)
THEN ADD to working memory the fact : (working-with ram)

R2 : IF (day monday)
THEN ADD to working memory the fact : (talking-to ram)

R3 : IF (talking-to X) AND (working-with X)
THEN ADD to working memory the fact : (busy-at-work X)

R4 : IF (busy-at-work X) OR (at-office X)
THEN ADD to working memory the fact : (not-at-home X)

R5 : IF (not-at-home X)
THEN DELETE from working memory the fact : (happy X)

R6 : IF (working-with X)
THEN DELETE from working memory the fact : (does-not-like X)

Now **to start the process of inference through forward chaining**, the rule based system will first search for all the rule/s whose antecedent part/s are satisfied by the current set of facts in the working memory. *For example, in this example, we can see that the rules R1 and R2 are satisfied, so the system will chose one of them using its* ***conflict resolution strategies.*** Let the rule R1 is chosen. So (working-with ram) is added to the working memory (after substituting "ram" in place of X). So working memory now looks like:

(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

Now this cycle begins again, the system looks for rules that are satisfied, it finds rule R2 and R6. Let the system chooses rule R2. So now (taking-to ram) is added to working memory. So now working memory contains following:

(talking-to ram)
(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

Now in the next cycle, rule R3 fires, so now (busy-at-work ram) is added to working memory, which now looks like:

(busy-at-work ram)
(talking-to ram)
(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

Now antecedent parts of rules R4 and R6 are satisfied. Let rule R4 fires, so (not-at-home, ram) is added to working memory which now looks like :

(not-at-home ram)
(busy-at-work ram)
(talking-to ram)

(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)
In the next cycle, rule R5 fires so (at-home ram) is removed from the working memory :

(not-at-home ram)
(busy-at-work ram)
(talking-to ram)
(working-with ram)
(day monday)
(does-not-like ram)

The forward chining will continue like this. But we have to be sure of one thing, that the ordering of the rules firing is important. A change in the ordering sequence of rules firing may result in a different working memory.

6.2.2 Backward Chaining Systems

In forward chining systems we have seen how rule-based systems are used to draw new conclusions from existing data and then add these conclusions to a working memory. **The forward chaining approach is most useful when** we know all the initial facts, but we don't have much idea what the conclusion might be.

If we know what the conclusion would be, or have some specific hypothesis to test, forward chaining systems may be inefficient. In forward chaining we keep on moving ahead until no more rules apply or we have added our hypothesis to the working memory. But in the process the system is likely to do a lot of additional and irrelevant work, adding uninteresting or irrelevant conclusions to working memory. Let us say that in the example discussed before, suppose we want to find out whether "ram is at home". We could repeatedly fire rules, updating the working memory, checking each time whether **`(at-home ram)`** is found in the new working memory. But maybe we had a whole batch of rules for drawing conclusions about what happens when I'm working, or what happens on Monday–we really don't care about this, so would rather only have to draw the conclusions that are relevant to the goal.

This can be done by ***backward chaining*** from the goal state or on some hypothesized state that we are interested in. This is essentially how Prolog works. Given a goal state to try and prove, for example ***(at-home ram),*** the system will first check to see if the goal matches the initial facts given. If it does, then that goal succeeds. If it doesn't the system will look for rules whose conclusions i.e., *actions* match the goal. One such rule will be chosen, and the system will then try to prove any facts in the preconditions of the rule using the same procedure, setting these as new goals to prove. **We should note that a backward chaining system does not need to update a working memory.** Instead it needs to keep track of what goals it needs to prove its main hypothesis. So we can say that **in a backward chaining system, the reasoning proceeds "backward", beginning with the goal to be established, chaining through rules, and finally anchoring in facts**.

Although, in principle same set of rules can be used for both forward and backward chaining. However, **in backward chaining, in practice we may choose to write the rules slightly differently**. In backward chaining we are concerned with matching the conclusion of a rule against some goal that we are trying to prove. So the 'then or consequent' part of the rule is usually not expressed as an action to take (e.g., add/delete), but as a state which will be true if the premises are true.

To learn more, let us take a different example in which we use backward chaining (The system is used to identify an animal based on its properties stored in the working memory):

Example

1. Let us assume that the working memory initially contains the following facts:

(has-hair raja)	representing	the fact "raja has hair"
(big-mouth raja)	representing	the fact "raja has a big mouth"
(long-pointed-teeth raja)	representing	the fact "raja has long pointed teeth"
(claws raja)	representing	the fact "raja has claws"

Let, the existing set of rules are:

1. IF (gives-milk X)
   THEN (mammal X)

2. IF (has-hair X)
   THEN (mammal X)

3. IF (mammal X) AND (eats-meat X)
   THEN (carnivorous X)

4. IF (mammal X) AND (long-pointed-teeth X) AND (claws X)
   THEN (carnivorous X)

5. IF (mammal X) AND (does-not-eat-meat X)
   THEN (herbivorous X)

6. IF (carnivorous X) AND (dark-spots X)
   THEN (cheetah, X)

7. IF (herbivorous X) AND (long-legs X) AND (long-neck X) AND (dark-spots X)
   THEN (giraffe, X)

8. IF (carnivorous X) AND (big-mouth X)
   THEN (lion, X)

9. IF (herbivorous X) AND (long-trunk X) AND (big-size X)
   THEN (elephant, X)

10. IF (herbivorous, X) AND (white-color X) AND ((black-strips X)
    THEN (zebra, X)

Now to start the process of inference through backward chaining, the rule based system will first form a hypothesis and then it will use the antecedent – consequent rules (previously called condition – action rules) to work backward toward hypothesis supporting assertions or facts.

Let us take the initial hypothesis that "raja is a lion" and then reason about whether this hypothesis is viable using backward chaining approach explained below :

- The system searches a rule, which has the initial hypothesis in the consequent part that someone i.e., raja is a lion, which it finds in rule 8.

- The system moves from consequent to antecedent part of rule 8 and it finds the first condition i.e., the first part of antecedent which says that "raja must be a carnivorous".

- Next the system searches for a rule whose consequent part declares that someone i.e., "raja is a carnivorous", two rules are found i.e., rule 3 and rule 4. We assume that the system tries rule 3 first.

- To satisfy the consequent part of rule 3 which now has become the system's new hypothesis, the system moves to the first part of antecedent which says that X i.e., raja has to be mammal.

- So a new sub-goal is created in which the system has to check that "raja is a mammal". It does so by hypothesizing it and tries to find a rule having a consequent that someone or X is a mammal. Again the system finds two rules, rule 1 and rule 2. Let us assume that the system tries rule 1 first.

- In rule 1, the system now moves to the first antecedent part which says that X i.e., raja must give milk for it to be a mammal. The system cannot tell this because this hypothesis is neither supported by any of the rules and also it is not found among the existing facts in the working memory. So the system abandons rule 1 and try to use rule 2 to establish that "raja is a mammal".

- In rule 2, it moves to the antecedent which says that X i.e., raja must have hair for it to be a mammal. The system already knows this as it is one of the facts in working memory. So the antecedent part of rule 2 is satisfied and so the consequent that "raja is a mammal" is established.

- Now the system backtracks to the rule 3 whose first antecedent part is satisfied. In second condition of antecedent if finds its new sub-goal and in turn forms a new hypothesis that X i.e., raja eats meat.

- The system tries to find a supporting rule or an assertion in the working memory which says that "raja eats meat" but it finds none. So the system abandons the rule 3 and try to use rule 4 to establish that "raja is carnivorous".

- In rule 4, the first part of antecedent says that raja must be a mammal for it to be carnivorous. The system already knows that "raja is a mammal" because it was already established when trying to satisfy the antecedents in rule 3.

- The system now moves to second part of antecedent in rule 4 and finds a new sub-goal in which the system must check that X i.e., raja has long-pointed-teeth which now becomes the new hypothesis. This is already established as " raja has long-pointed-teeth" is one of the assertions of the working memory.

- In third part of antecedent in rule 4 the system's new hypothesis is that "raja has claws". This also is already established because it is also one the assertions in the working memory.

- Now as all the parts of the antecedent in rule 4 are established so its consequent i.e., "raja is carnivorous" is established.

- The system now backtracks to rule 8 where in the second part of the antecedent says that X i.e., raja must have a big-mouth which now becomes the new hypothesis. This is already established because the system has an assertion that "raja has a big mouth".

- Now as the whole antecedent of rule 8 is satisfied **so the system concludes that "raja is a lion".**

We have seen that the system was able to work backward through the antecedent – consequent rules, using desired conclusions to decide that what assertions it should look for and ultimately establishing the initial hypothesis.

How to choose the type of chaining among forward or backward chaining for a given problem ?

Many of the rule based deduction systems can chain either forward or backward, but which of these approaches is better for a given problem is the point of discussion.

First, let us learn some basic things about rules i.e., **how a rule relates its input/s (i.e., facts) to output/s (i.e., conclusion)**. Whenever in a rule, a particular set of facts can lead to many conclusions, the rule is said to have a high degree of **<u>fan out</u>**, and a strong candidate of backward chaining for its processing. On the other hand, whenever the rules are such that a particular hypothesis can lead to many questions for the hypothesis to be established, the rule is said to have a high degree of fan in, and a high degree of **<u>fan in</u>** is a strong candidate of forward chaining.

To summarize, the following points should help in choosing the type of chaining for reasoning purpose :

- If the set of facts, either we already have or we may establish, can lead to a large number of conclusions or outputs , but the number of ways or input paths to reach that particular conclusion in which we are interested is small, then **the degree of fan out is more than degree of fan in. In such case, backward chaining is the preferred choice.**
- But, if the number of ways or input paths to reach the particular conclusion in which we are interested is large, but the number of conclusions that we can reach using the facts through that rule is small, then **the degree of fan in is more than the degree of fan out. In such case, forward chaining is the preferred choice**.

For case where **the degree of fan out and fan in are approximately same**, then in case if not many facts are available and the problem is check if one of the many possible conclusions is true**, backward chaining is the preferred choice**.

6.2.3 Conflict Resolution

Next, we discuss in detail some of the issues involved in a rule-based system.

Rule-based systems vary greatly in their details and syntax, A basic principle of rule-based system is that each rule is an independent piece of knowledge. In an IF-THEN rule, the IF-part contains all the conditions for the application of the rule under consideration. THEN-part tells the action to be taken by the interpreter. The interpreter need not search any where else except within the rule itself for the conditions required for application of the rule.

Another important consequence of the above-mentioned characteristic of a rule-based system is that no rule can call upon any other and hence rules are ignorant and hence independent, of each other. This gives a highly modular structure to the rule-based systems. Because of the highly modular structure of the rule-base, the rule-based system addition, deletion and modification of a rule can be done without any danger side effects.

The main problem with the rule-based systems is that when the rule-base grows and becomes very large, then checking (i) whether a new rule intended to be added is redundant, i.e., it is already covered by some of the earlier rules. Still worse, as the rule- base grows, checking the consistency of the rule-base also becomes quite difficult. By consistency, we mean there may be two rules having similar conditions, the actions by the two rules conflict with each other.

Some of the <u>conflict resolution strategies</u> which are used to decide which rule to fire are given below:

- Don't fire a rule twice on the same data.
- Fire rules on more recent working memory elements before older ones. This allows the system to follow through a single chain of reasoning, rather than keeping on drawing new conclusions from old data.
- Fire rules with more specific preconditions before ones with more general preconditions. This allows us to deal with non-standard cases.

These strategies may help in getting reasonable behavior from a forward chaining system, but **the most important thing is how should we write the rules**. They should be carefully constructed, with the preconditions specifying as precisely as possible when different rules should fire. Otherwise we will have little idea or control of what will happen.

To understand, let us take the following example in which we use forward chaining:

Example

Let us assume that the working memory initially contains the following facts :

(day monday)
(at-home ram)
(does-not-like ram)

Let, the existing set of rules are:

R1 : IF (day monday)
THEN ADD to working memory the fact : (working-with ram)

R2 : IF (day monday)
THEN ADD to working memory the fact : (talking-to ram)

R3 : IF (talking-to X) AND (working-with X)
THEN ADD to working memory the fact : (busy-at-work X)

R4 : IF (busy-at-work X) OR (at-office X)
THEN ADD to working memory the fact : (not-at-home X)

R5 : IF (not-at-home X)
THEN DELETE from working memory the fact : (happy X)

R6 : IF (working-with X)
THEN DELETE from working memory the fact : (does-not-like X)

Now **to start the process of inference through forward chaining**, the rule based system will first search for all the rule/s whose antecedent part/s are satisfied by the current set of facts in the working memory. *For example, in this example, we can see that the rules R1 and R2 are satisfied, so the system will chose one of them using its* ***conflict resolution strategies***. Let the rule R1 is chosen. So (working-with ram) is added to the working memory (after substituting "ram" in place of X). So working memory now looks like:

(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

Now this cycle begins again, the system looks for rules that are satisfied, it finds rule R2 and R6. Let the system chooses rule R2. So now (taking-to ram) is added to working memory. So now working memory contains following:

(talking-to ram)
(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

Now in the next cycle, rule R3 fires, so now (busy-at-work ram) is added to working memory, which now looks like:

(busy-at-work ram)
(talking-to ram)
(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

Now antecedent parts of rules R4 and R6 are satisfied. Let rule R4 fires, so (not-at-home, ram) is added to working memory which now looks like :

(not-at-home ram)
(busy-at-work ram)
(talking-to ram)
(working-with ram)
(day monday)
(at-home ram)
(does-not-like ram)

In the next cycle, rule R5 fires so (at-home ram) is removed from the working memory :

(not-at-home ram)
(busy-at-work ram)

(talking-to ram)
(working-with ram)
(day monday)
(does-not-like ram)

The forward chining will continue like this. But we have to be sure of one thing, that the ordering of the rules firing is important. A change in the ordering sequence of rules firing may result in a different working memory.

Check your Progress - 1

Exercise 1 ; In the "Animal Identifier System" discussed above use forward chaining to try to identify the animal called "raja".

6.3 SEMANTIC NETS

Semantic Network representations provide a **structured knowledge representation**. In such a network, parts of knowledge are clustered into semantic groups. In semantic networks, the concepts and entities/objects of the problem domain are represented by nodes and relationships between these entities are shown by arrows, generally, by directed arrows. In view of the fact that semantic network **representation is a pictorial depiction** of objects, their attributes and the relationships that exist between these objects and other entities. A semantic net is just a graph, where the nodes in the graph represent concepts, and the arcs are labeled and represent binary relationships between concepts. These networks provide a more natural way, as compared to other representation schemes, for mapping to and from a natural language.

For example, the fact (a piece of knowledge): ***Mohan struck Nita in the garden with a sharp knife last week***, is represented by the semantic network shown in *Figure 1.1*.

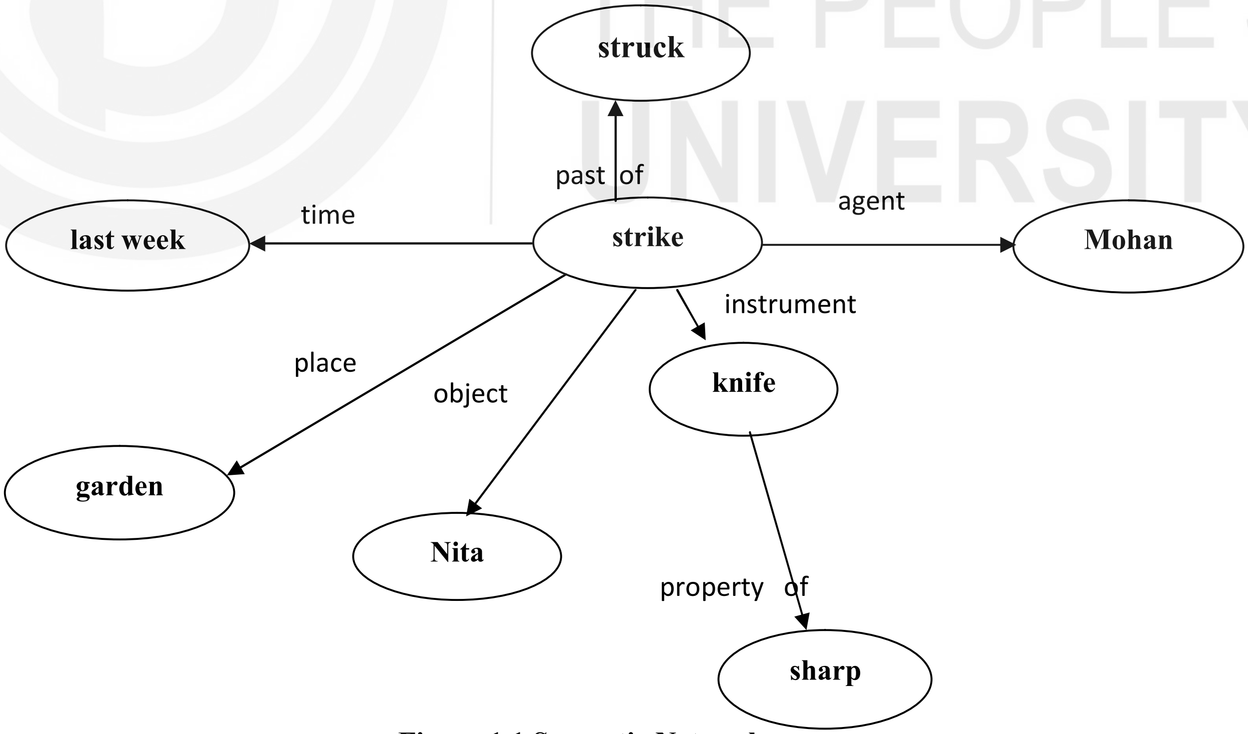


Figure 1.1 Semantic Network

The two most important relations between concepts are *(i) subclass* relation between a class and its superclass, and *(ii) instance* relation between an object and its class. Other relations may be *has-part*, *color* etc. As mentioned earlier, relations are indicated by labeled arcs.

As information in semantic networks is clustered together through relational links, the knowledge required for the performance of some task is generally available within short spatial span of the semantic network. This type of knowledge organisation in some way, resembles the way knowledge is stored and retrieved by human beings.

Subclass and *instance* relations allow us to use ***inheritance*** to infer new facts/relations from the explicitly represented ones. We have already mentioned that **the graphical portrayal of knowledge in semantic networks,** being visual, is easier than other representation schemes for the human beings to comprehend. This fact helps the human beings to guide the expert system, whenever required. This is perhaps the reason for the popularity of semantic networks.

Check Your Progress – 2
Exercise 2: Draw a semantic network for the following English statement:
Mohan struck Nita and Nita's mother struck Mohan.

6.4 FRAMES

Frames are a variant of semantic networks that are one of the popular ways of representing non-procedural knowledge in an expert system. In a frame, all the information relevant to a particular concept is stored in a single complex entity, called a frame. Frames look like the data structure, record. Frames support inheritance. They are often used to capture knowledge about *typical* objects or events, such as a car, or even a mathematical object like rectangle. As mentioned earlier, a frame is a structured object and different names like *Schema*, *Script*, *Prototype*, and even *Object* are used in stead of frame, in computer science literature.

We may represent some knowledge about a lion in frames as follows:

Mammal :
  Subclass : Animal
  warm_blooded : yes

Lion :
  subclass : Mammal
  eating-habbit : carnivorous
  size : medium

Raja :
  instance : Lion
  colour : dull-Yellow
  owner : Amar Circus

Sheru :
  instance : Lion
  size : small

A particular frame (such as Lion) has a number of *attributes* or *slots* such as *eating-habit* and *size*. Each of these slots may be filled with particular values, such as the *eating-habit* for lion may be filled up as *carnivorous*.

Sometimes a slot contains additional information such as how to apply or use the slot values. Typically, a slot contains information such as *(attribute, value)* pairs, default values, conditions for filling a **slot**, pointers to other related frames, and also procedures that are activated when needed for different purposes.

In the case of frame representation of knowledge, **inheritance is simple** if an object has a single parent class, and if each slot takes a single value. For example, if a mammal is warm blooded then automatically a lion being a mammal will also be warm blooded.

But **in case of multiple inheritance** i.e., in case of an object having more than one parent class, we have to decide which parent to inherit from. For example, a lion may inherit from "wild animals" or "circus animals". In general, both the slots and slot values may themselves be frames and so on.

Frame systems are pretty complex and sophisticated knowledge representation tools. This representation has become so popular that special high level frame based representation languages have been developed. Most of these languages use LISP as the host language. It is also possible to represent frame-like structures using object oriented programming languages, extensions to the programming language LISP.

Check Your Progress – 3
Exercise 3: Define a frame for the entity *date* which consists of *day*, *month* and *year*. each of which is a number with restrictions which are well-known. Also a procedure named *compute-day-of-week* is already defined.

6.5 SCRIPTS

A script is a structured representation describing a stereotyped sequence of events in a particular context.

Scripts are used in natural language understanding systems to organize a knowledge base in terms of the situations that the system should understand. Scripts use a frame-like structure to represent the commonly occurring experience like going to the movies eating in a restaurant, shopping in a supermarket, or visiting an ophthalmologist.

Thus, a script is a structure that prescribes a set of circumstances that could be expected to follow on from one another.

Scripts are beneficial because:

- Events tend to occur in known runs or patterns.
- A casual relationship between events exist.
- An entry condition exists which allows an event to take place.
- Prerequisites exist upon events taking place.

Components of a script

The components of a script include:

- **Entry condition:** These are basic condition which must be fulfilled before events in the script can occur.
- **Results:** Condition that will be true after events in script occurred.
- **Props:** Slots representing objects involved in events
- **Roles:** These are the actions that the individual participants perform.
- **Track:** Variations on the script. Different tracks may share components of the same scripts.
- **Scenes:** The sequence of events that occur.

Describing a script, special symbols of actions are used. These are:

Symbol	Meaning	Example
ATRANS	transfer a relationship	give
PTRANS	transfer physical location of an object	go
PROPEL	apply physical force to an object	push
MOVE	move body part by owner	kick
GRASP	grab an object by an actor	hold
INGEST	taking an object by an animal eat	drink
EXPEL	expel from animal's body	cry
MTRANS	transfer mental information	tell
MBUILD	mentally make new information	decide
CONC	conceptualize or think about an idea	think
SPEAK	produce sound	Say
ATTEND	focus sense organ	listen

Example:-Script for going to the bank to withdraw money.

SCRIPT : Withdraw money

TRACK : Bank

PROPS : Money

Counter

Form

Token

Roles :

P= Customer

E= Employee

C= Cashier

Entry conditions: P has no or less money.

The bank is open.

Results : P has more money.

Scene 1: Entering

P PTRANS P into the Bank

P ATTEND eyes to E

P MOVE P to E

Scene 2: Filling form

P MTRANS signal to E

E ATRANS form to P

P PROPEL form for writing

P ATRANS form to P

ATRANS form to P

Scene 3: Withdrawing money

P ATTEND eyes to counter

P PTRANS P to queue at the counter

P PTRANS token to C

C ATRANS money to P

Scene 4: Exiting the bank

P PTRANS P to out of bank

Advantages of Scripts

- Ability to predict events.
- A single coherent interpretation maybe builds up from a collection of observations.

Disadvantages of Scripts

- Less general than frames.
- May not be suitable to represent all kinds of knowledge

6.6 SUMMARY

This unit majorly discussed the various knowledge representation mechanisms, used in Artificial Intelligence. The unit begins with the discussion on Rule Based Systems, and discussed the related concept of Forward chaining and Backward chaining, later the concept of Conflict resolution is discussed. The unit also discussed the other techniques of knowledge representation like Semantic nets, Frames and Scripts; along with relevant examples for each.

6.7 SOLUTIONS/ANSWERS

Check Your Progress – 1

Exercise 1: Refer to section 6.2

Check Your Progress – 2

Exercise 2: Refer to section 6.3

Check Your Progress – 3

Exercise 3: Refer to section 6.4

6.8 FURTHER READINGS

1. Ela Kumar, “ Artificial Intelligence”, IK International Publications
2. E. Rich and K. Knight, “Artificial intelligence”, Tata Mc Graw Hill Publications
3. N.J. Nilsson, “Principles of AI”, Narosa Publ. House Publications
4. John J. Craig, “Introduction to Robotics”, Addison Wesley publication
5. D.W. Patterson, “Introduction to AI and Expert Systems" Pearson publication

UNIT 7 PROBABILISTIC REASONING

Structure **Page Nos.**

7.0 INTRODUCTION

This unit is dedicated to probability theory and its usage in decision making for various problems. Contrary to the classical decision making of True and False propositions, the probability of the truth value with a certain probability is used for making decisions. The inclusion of such a probabilistic approach is quite relevant since uncertainties are quite obvious in the real world.

As we know, the probability of an event (uncertain event I) is basically the measure of the degree of likelihood of the occurrence of event I. Let the set of all such possible events is represented as sample space S The measure of probability is a function *P ()* mapping the event outcome E_i from sample space S to some real number and satisfying few conditions such as:

(i) $0 \leq P(I) \leq 1$ for any event $I \subseteq S$

(ii) $P(S) = 1$, represents a certain outcome, and

(iii) For $E_i \cap E_j = \phi$, for all $i \neq j$ (the E_i are mutually exclusive), i.e. $P(E_1 \cup E_2 \dots) = P(E_1) + P(E_2) + \dots$

Using the above mentioned three conditions, we can derive the basic laws of probability. It is also to be noted that only these three conditions are not enough to compute the probability of an outcome. This additionally requires the collection of experimental data for estimating the underlying distribution.

7.1 OBJECTIVES

After going through this unit, you should be able to:

- Understand the role of probabilistic reasoning in AI
- Understand the Concept of Bayesian theory and Bayesian networks
- Perform probabilistic inference through Bayesian Networks
- Understand the other Paradigm of Uncertain Reasoning & Dempster Scheffer Theory

7.2 REASONING WITH UNCERTAIN INFORMATION

Reasoning is an important step for various decision making. The amount of information and its correctness plays a crucial role in reasoning. Decision making is easier when we have certain information i.e. the correctness of information can be ascertained. In the other situation when the certainty of information can not be ascertained, the decision-making process is likely to be erroneous or may not be correct. In this situation how decisions are made with some uncertainty (uncertain information) is the core objective of this unit. If we talk about the sources of uncertainty in the information, this could be due to various reasons including experimental error, instrument fault, unreliable source and any other reason. Once the information is received and we have to make decisions based on received uncertain information, we can not rely on models which use certain information. One of the potential solutions appears to be probabilistic reasoning for such scenarios. We can make use of probabilistic models for reasoning with uncertain information with some probability. Let’s first see the basic probability concepts before discussing probabilistic reasoning.

7.3 REVIEW OF PROBABILITY THEORY

Now, you are familiar with the reasoning and how it can be useful with probability theory. Before we dive deeper into the Bayes’ theorem and its applications, let us review some of the basic concepts of probability theory. These concepts will be helping us to understand other topics of this unit.

Trials, Sample Space, Events : You must have often observed that a random experiment may comprise a series of smaller sub-experiments. These are called trials. Consider for instance the following situations.

Example 1: Suppose the experiment consists of observing the results of three successive tosses of a coin. Each toss is a trial and the experiment consist of three trials so that it is completed only after the third toss (trial) is over.

Example 2: Suppose from a lot of manufactured items, ten items are chosen successively following a certain mechanism for checking. The underlying experiment is completed only after the selection of the tenth item is completed; the experiment obviously comprises 10 trials.

Example 3: If you consider Example 1 once again you would notice that each toss (trial) results into either a head (H) or a tail (T). In all there are 8 possible outcomes of the experiment viz., s_1 = (H,H,H), s_2 = (H,H,T), s_3 = (H,T,H), s_4 = (T,H,H), s_5 =(1,T,H), s_6 = (T,H,T), s_7 =(H,T,T) and s_8 = (T, T, T).

Let ζ be a fixed sample space. We have already defined an event as a collection of sample points from ζ. Imagine that the (conceptual) experiment underlying ζ is being performed. The phrase "the event E occurs" would mean that the experiment results in an outcome that is included in the event E. Similarly, non-occurrence of the event E would mean that the experiment results into an outcome that is not an element of the event E. Thus, the collection of all sample points that are not included in the event E is also an event which is complementary to E and is denoted as E^c. The event E^c is therefore the event which contains all those sample points of ζ which are not in E. As such, it is easy to see that the event E occurs if and only if the event E^c does not take place. The events E and E^c are complementary events and taken together they comprise the entire sample space, i.e., $E \; E^c = \zeta$.

You may recall that ζ is an event which consists of all the sample points. Hence, its complement is an empty set in the sense that it does not contain any sample point and is called the null event, usually denoted as ø so that $\zeta^c = ø$.

Let us once again consider Example 3. Consider the event E that the three tosses produce at least one head. Thus, $E = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7\}$ so that the complementary event $E^c=\{s_8\}$, which is the event of not scoring a head at all. Again in Example 3 in the case of selection without replacement, event that the white marble is picked up at least once is defined as $E = \{(r_1,w), (r_2,w), (w, r_2) (w, r_1)\}$. Hence, $E^c = \{(r_1, r_2), (r_2 , r_1)\}$ i.e. the event of not picking the white marble at all.

Let us now consider two events E and F. We write E F, read as E "union" F, to denote the collection of sample points, which are responsible for occurrence of either E or F or both. Thus, E F is a new event and it occurs if and only if either E or F or both occur i.e. if and only if at least one of the events E or F occurs. Generalizing this idea, we can define a new event Ej, read as "union" of the k events $E_1, E_2,..., E_k$, as the event which consists of all sample points that are in at least one of the events $E_1, E_2, \ldots E_k$ and it occurs if and only if at least one of the events $E_1, E_2,...,E_k$ occurs.

Again, let E and F be two given events. We write $E \cap F$, read as E "Intersection" F, to denote the collection of sample points any of whose occurrence implies the occurrence of both E and F. Thus, $E \cap F$ is a new event and it occurs if and only if both the events E and F occur. Generalizing this idea, we can define a new event Ej read as "intersection" of the k events $E_1, E_2,...,E_k$, as the event which consists of sample points that are common to each of the events $E_1, E_2,..., E_k$, and it occurs only if all the k events $E_1, E_2,...,E_k$ occur simultaneously. Further, two events E and F are said to be mutually exclusive or disjoint if they do not have a common sample point i.e. $E \cap F = ø$.

Two mutually exclusive events then cannot occur simultaneously. In the coin-tossing experiment for instance, the two events, heads and tails, are mutually exclusive: if one occurs, the other cannot occur. To have a better understanding of these events let us once again look at Example 3. Let E be the event of scoring an odd number of heads and F be the event that tail appears in the first two tosses, so that $E = \{s_1, s_5, s_6, s_7\}$ and $F = \{s_5, s_8\}$. Now $E \cap F = \{s_5\}$, the event that only the third toss yields a head. Thus events E and F are not mutually exclusive.

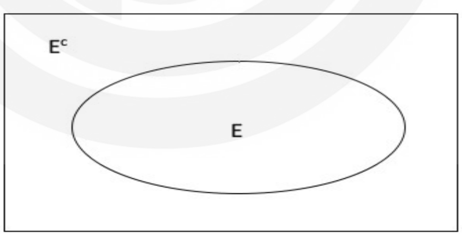


Fig. 1(a)

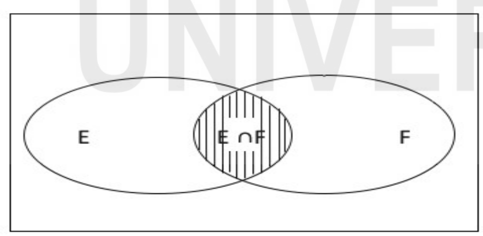


Fig.1(b)

The above relations between events can be best viewed through a Venn diagram. A rectangle is drawn to represent the sample space ζ. All the sample points are represented within the rectangle by means of points. An event is represented by the region enclosed by a closed curve containing all the sample points leading to that event. The space inside the rectangle but outside the closed curve representing E represents the complementary event E^c (See Fig.1(a) above) Similarly, in Fig.1(b), the space inside the curve represented by the broken line represent the event E U F and the shaded portion represents $E \cap F$.

As is clear by now, the outcome of a random experiment being uncertain, none of the various events associated with a sample space can be predicted with certainty before the underlying experiment is performed and the outcome of it is noted. However, some events may intuitively seem to be more likely than the rest. For example, talking about human beings, the event that a person will live 20 years seems to be more likely compared to the event that the person will live 200 years. Such thoughts motivate us to explore if one can construct a scale of measurement to distinguish between likelihoods of various events. Towards this, a small but extremely significant fact comes to our help. Before we elaborate on this, we need a couple of definitions.

Consider an event E associated with a random experiment; suppose the experiment is repeated n times under identical conditions and suppose the event E (which is not likely to occur with every performance of the experiment) occurs $f_n(E)$ times in these n repetitions. Then, $f_n(E)$ is called the frequency of the event E in n repetitions of the experiment and $r_n.(E) = f_n,(E)/n$ is called the relative frequency of the event E in n repetitions of the experiment. Let us consider the following example.

Example 4: Consider the experiment of throwing a coin. Suppose we repeat the process of throwing a coin 5 times and suppose the frequencies of occurrence of head is tabulated below in Table-1:

No. of repetitions (n)	Frequency of head (f_n(H)	Relative frequency of head r_n(H)
1	0	0
2	1	1/2
3	2	2/3
4	3	3/4
5	3	3/5

Notice that the third column in Table-1 gives the relative frequencies r_n (H) of heads. We can keep on increasing the number of repetitions n and continue calculating the values of r_n (H) in Table 1. Merely to fix ideas regarding the concept of probability of an event, we present below a very naive approach which in no way is rigorous, but it helps to see things better at this stage.

Check Your Progress- 1

Problem -1. In each of the following exercises, an experiment is described. Specify the relevant sample spaces:

a) A machine manufactures a certain item. An item produced by the machine is tested to determine whether or not it is defective.

b) An urn contains six balls, which are colored differently. A ball is drawn from the urn and its color is noted.

c) An urn contains ten cards numbered 1 through 10. A card is drawn, its number noted and the card is replaced. Another card is drawn and its number is noted.

Problem 2. Suppose a six-faced die is thrown twice. Describe each of the following events:

i) The maximum score is 6.
ii) The total score is 9.
iii) Each throw results in an even score.
iv) Each throw results in an even score larger than 2.
v) The scores on the two throws differ by at least 2.

7.3.1 Conditional probability and independent events

Let ζ be the sample space corresponding to an experiment and E and F are two events of ζ Suppose the experiment is performed and the outcome is known only partially to the effect that the event F has taken place. Thus there still remains a scope for speculation about the occurrence of the other event E. Keeping this additional piece of information confirming the occurrence of F in view, it would be appropriate to modify the probability of occurrence of E suitably. That such modifications would be necessary can be readily appreciated through two simple instances as follows:

Example 5: Suppose, E and F are such that F E so that occurrence of F would automatically imply the occurrence of E. Thus with the information that the event F has taken place in view, it is plausible to assign probability 1 to the occurrence of E irrespective of its original probability.

Example 6: Suppose. E and F are two mutually exclusive events and thus they cannot occur together. Thus whenever we come to know that the event F has taken place, we can rule out the occurrence of E. Therefore, in such a situation, it will be appropriate to assign probability 0 to the occurrence of E.

Example 7: Suppose a pair of balanced dice A and B are rolled simultaneously so that each of the 36 possible outcomes is equally likely to occur and hence has probability   Let E be the event that the sum of the two scores is 10 or more and F be the event that exactly one of the two scores is 5.

Then E = {(4.6), (5.5), (5,6), (6,4), (6,5), (6,6)} so that P(E) = 6/36 = 1/6.

Also, F= {(1.5), (2,5), (3,5), (4,5), (6,5), (5,1), (5,2), (5,3), (5,4), (5,6)}.

Now suppose we are told that the event F has taken place (note that this is only partial information relating to the outcome of the experiment). Since each of the outcome originally had the same probability of occurring, they should still have equal probabilities. Thus given that exactly one of the two scores is 5 each of the 10 outcomes of event F has probability while the probability of remaining 26 points in the sample space is 0.

In the light of the information that the event F has taken place the sample points (4,6), (6,4), (5,5) and (6,6) in the event E must not have materialized. One of the two sample points (5,6) or (6,5) must have materialized. Therefore the probability of E would no longer be 1/6. Since all the 10 sample points in F are equally likely, the revised probability of E given the occurrence of F, which occurs through the materialization of one of the two sample points (6,5) or (5,6) should be 2/10 = 1/5.

The probability just obtained is called the conditional probability that E occurs given that F has occurred and is denoted by P(E|F). We shall now derive a general formula for calculating P(E|F).

Consider the following probability table:

Table 2

Events	E	E^c
F	P	Q
F^c	r	s

In Table 2, $P(E \cap F) = p$, $P(E^c \cap F) = q$, $P(E \cap F^c) = r$ and $P(E^c \cap F^c) = s$ and hence, $P(E)=P(E \cap F) \cup (E \cap F^c)) = P(E \cap F) + P(E \cap F^c) = p+r$ and similarly, $P(F) = q + s$.

Now suppose that the underlying random experiment is being repeated a large number of times, say N times. Thus, taking a cue from the long term relative frequency interpretation of probability, the approximate number of times the event F is expected to take place will be NP(F) = N(q+s). Under the condition that the event F has taken place, the number of times the event E is expected to take place would be NP(E ∩ F) as both E and F must occur simultaneously. Thus, the long term relative frequency of E under the condition of occurrence of F, i.e. the probability of occurrence of E under the condition of occurrence of F, should be NP(E ∩ F)/NP(F) = P(E ∩ F)/P(F). This is the proportion of times E occurs out of the repetitions where F takes place. With the above background, we are now ready to define formally the conditional probability of an event given another.

Definition: Let E and F be two events from a sample space ζ. The conditional probability of the event E given the event F, denoted by P(E|F), is defined as P(E|F) = P(E ∩ F)/P(F), whenever P(F) > 0.

When P(F) = 0, we say that P(E|F) is undefined. We can also write from Eqn. P(E ∩ F) = P(E|F)P(F).

Referring back to Example 3, we see that P(E) = 6/36,P(F) = 10/36; since, E ∩ F= {(5,6), (6,5)}, P(E ∩ F) = 2/36, P(E|F) = (2/36)/(10/36) = 2/10 = 1/5, which is the same as that obtained in Example 3. Another result can be generalized to k events E_1 E_2, ..., E_k, where k $\geq$2. And now an exercise for you.

Check Your Progress 2

Problem-1: In a class, three students tossed one coins (one each) for 3 times. Write down all the possible outcomes which can be obtained in this experiment.

Problem-2: In problem 1, what is the probability of getting 2 more than 2 heads at a time. Also write the probability of getting three tails at a time.

Problem-3: In problem 1 calculate the Relative frequency of tail r_n(T).

7.4 INTRODUCTION TO BAYESIAN THEORY

Bayes' theorem is widely used to calculate the conditional probabilities of events without a joint probability. It is also used to calculate the conditional probability where intuition fails. In simple terms, probability of a given hypothesis H conditional on E can be defined as P_E (H) = P(H \& E)/ P(E), where P(E) > 0, and the term P(H \& E) also exists. Here P_E is referred to as a probability function. To simply understand the Bayes' theorem, have a look at the following definitions.

Joint Probability: This refers to the probability of two or more events simultaneously occurring, e.g. P(A and B) or P(A, B).

Marginal Probability: It is the probability of an event occurring irrespective of outcome of the other random variables e.g. P(A).

Conditional probability: A conditional probability is defined as the probability of occurrence of an event provided that another event has occurred. e.g. P(A | B).

The conditional probability can also be written in terms of joint probability as P(A|B) = P(A, B)/ P(B). In other way, if one conditional probability is given, other can be calculated as P(A|B) = P(B|A)*P(A) /P(B).

Let ‘S’ be a sample space in consideration. Let events ‘A1’ , ‘A2’ , ... ‘An’ is the set of mutually exclusive events in sample space ‘S’. Let ‘B’ be an event from sample space ‘S’ provided P(B) > 0, then according to Bayes’ theorem.

$P(A_k \mid B) = P(A_k \cap B) / P(A_1 \cap B)+P(A_2 \cap B) \ldots\ldots\ldots\ldots P(A_n \cap B)$, this can also be written in terms of Bayes’ theorem.

$P(A_k \mid B) = P(A_k)P(B \mid A_k) \;/\; P(A_1).P(B \mid A_1)+P(A_2).P(B \mid A_2) \ldots\ldots\ldots\ldots P(A_n).P(B \mid A_n)$

7.5 BAYE’S NETWORKS

The probabilistic models are being used in defining the relationships among variables and are used to calculate probabilities.The Bayes’ network is a simpler form of applying Bayes’ theorem to complex real world problems. This uses a probabilistic graphical model which captures the conditional dependence explicitly and is represented using directed edges in a graph. Here if we take fully conditional models, we may need a big amount of data to address all possible events/ cases and in such scenario probabilities may not be calculated practically. On the other hand, simple assumptions like conditional independence of random variables may turn out to be effective, giving a way for Bayes’ Network.

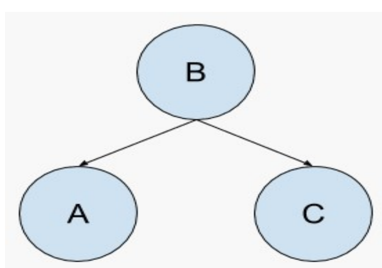
While representing a Bayes’ Network graphically, nodes represent the distribution of probabilities for random variables. The edges in the graph represent the relationship among random variables. The key benefits of a Bayes’ Network are model visualization, relationships among random variables and computations of complex probabilities.

Example 8: Let us now create a Bayesian Network for an example problem. Let us consider three random variables A, B and C. It is given that A is dependent on B, and C is dependent on B. The conditional dependence can be stated as P(A|B|) and P(C|B) for both the given statements respectively. Similarly the conditional independence can be stated here as P(A|B, C) and P(C|B, A) for both statements respectively.

Here, we can also write P(A|C, B) = P(A|B) as A is unaffected by the C. Also the joint probability of A and C given B can be written as product of conditional probabilities as P(A, C|B) = P(A|B) * P(C|B).

Now using Bayes' theorem, the joint probability of P(A, B, C) can be written as P(A,B,C)= P(A|B)*P(C|B)*P(B).

The corresponding graph is shown below in figure 1. Here each random variable is represented as a node and edges between nodes are conditional probabilities.



7.6 PROBABILISTIC INFERENCE

The probabilistic inference is very much dependent on the conditional probability of the specified events provided the information of occurrence of other events is available. For example, two events E and F such that P(F)>0, the conditional probability of event E when F has occurred can be written as :

$$P(E/F) = \frac{P(E \cap F)}{P(F)}$$

When an experiment is repeated a large number of times (say n), the above expression can be given a frequency interpretation. Let the number of occurrences of an event F is represented as No. (F) and the probability of a joint event of E and F as No. (E ∩ F). The relative frequencies of both these events can be computed as f_r:

$f_r(E \cap F) = \frac{No.(E \cap F)}{n}$ and similarly,

$$f_r(F) = \frac{No.(F)}{n}$$

Here, if n is large, the ratio of above two expressions represent the proportion of times the event E occurs relative to the occurrence of F. This can also be understood as the approximate conditional occurrence of event F with E.

$$f_r(E \cap F) / f_r(F) \simeq P(E \cap F) / P(F)$$

We can also write the conditional probability of event F while it is given that event E has already occurred, as

$$P(E / F) = P(E \cap F) / P(F)$$

Using above two equations we can also write

$$P(F / E) = P(E / F) P(F) / P(E)$$

The above expression is also one form of Bayes' Rule. Here the notion is simple: the probability of an event F occurring when we know the probability of an event E which has already occurred is the same as the probability of occurring of event E when the probability of occurrence of event F is known.

7.7 BASIC IDEA OF INFERENCING WITH BAYE'S NETWORKS

We are now aware of the Bayes theorem, probability and Bayes networks. Let's now talk about how inferences can be made using Bayes networks.A network here represents the degree of belief of proposition and their causal interdependence. The inference in a network can be done by propagating the given probabilities of related information through the network giving the output to one of the conclusion nodes. The network representation also reduces the time and space requirements for huge computations involving the probabilities of uncertain knowledge of propositional variables. Further, one can not make the inference from such a large data in real time. The solution to such a problem can be found using the network representation. Here the network of nodes represents variables connected by edges which represents causal influences (dependencies) among nodes. Here the edge weights can be used to represent the strength of influences or in other terms the conditional probabilities.

To use this type of probabilistic inference model, one first needs to assign probabilities to all basic facts in the underlying knowledge base. This requires the definition of an appropriate sample space and the assignment of a priori and conditional probabilities. In addition to this some methods must be selected to compute the combined probabilities when pooling evidence in a sequence of inference steps. In the end, when the outcome of an inference chain results in one or more proposed conclusions, the alternatives must be compared and one should be chosen on the basis of likelihood.

7.8 OTHER PARADIGM OF UNCERTAIN REASONING

The other ways of dealing with uncertainty are the ones with no theoretical proof. These are mostly based on intuition. These are selected over formal methods as a pragmatic solution to a particular problem, when the formal methods impose difficult or impossible conditions. One such ad hoc procedure is used to diagnose meningitis and infectious blood disease, the system is called MYCIN. The MYCIN uses If and then rules to assess various forms of patient evidence. It also measures both belief and disbelief to represent degree of confirmation and disconfirmation respectively in a given hypothesis. The ad hoc methods have been used in a larger number of knowledge-based systems than formal methods. This is due to the difficulties encountered in acquiring a large number of reliable probabilities related to the given domain and to the complexities to the ensuing calculations.

One other paradigm is to use Heuristic reasoning methods. These are based on the use of procedures, rules and other forms of encoded knowledge to achieve specified goals under certainty. Using both domain specific and general heuristics, one of several alternative conclusions may be chosen through the strength of positive vs negative evidence presented in the form of justification or endorsement.

The in depth and detailed discussion on this is not in the scope of this unit/ course.

7.9 DEMPSTER SCHEFFER THEORY

Let us now discuss a mathematical theory based only on the evidence, known as Dempster-Schafer (D-S) theory given by Dempster and extended by Shafer in "Mathematical Theory of Evidences". This uses a belief function to combine separate and independent evidence pieces to quantify the belief in a statement. The D-S theory is a generalization of Bayesian probability theory where multiple possible events are assigned probabilities opposed to mutually exclusive singletons. The D-S theory assumes the existence of ignorance in knowledge creating uncertainty which in turn induces belief. Here, uncertainty of the hypothesis is represented by the belief function. The main characteristic of the theory is:

1. Multiple possible events are permitted to assign probabilities.
2. These events should be exhaustive and exclusive.

Here, the multiple sources of information are assigned some degree of belief and then aggregated using the D-S combination rule. This also limits the theory for intensive computation because of the lack of independent assumptions from such a large number of information sources.

Let us now define a few terms used in D-S theory which will be useful for us.

7.9.1 Evidence

These are events related to one hypothesis or set of hypotheses. Hare, a relation is not permitted between various pieces of evidence or set of hypotheses. Also, the relation between the set of hypotheses and the piece of evidence is only quantified by a source of data. In context of D-S theory, we have four types of evidences as following:

a) Consonant Evidence: These are basically appearing in a nested structure where each subset is included into the next bigger subset and so on. Here with each increasing subset size, the information refines the evidentiary set over the time.
b) Consistent Evidence: This assures the presence of at least one common element to all the subsets.
c) Arbitrary Evidence: A situation where there is not a common element occurring in the subsets though some of the subsets may have a few common element(s).
d) Disjoint Evidence: There is no subset having common elements.

All these four evidence types can be understood by looking at the below given figure 2.(a-d).

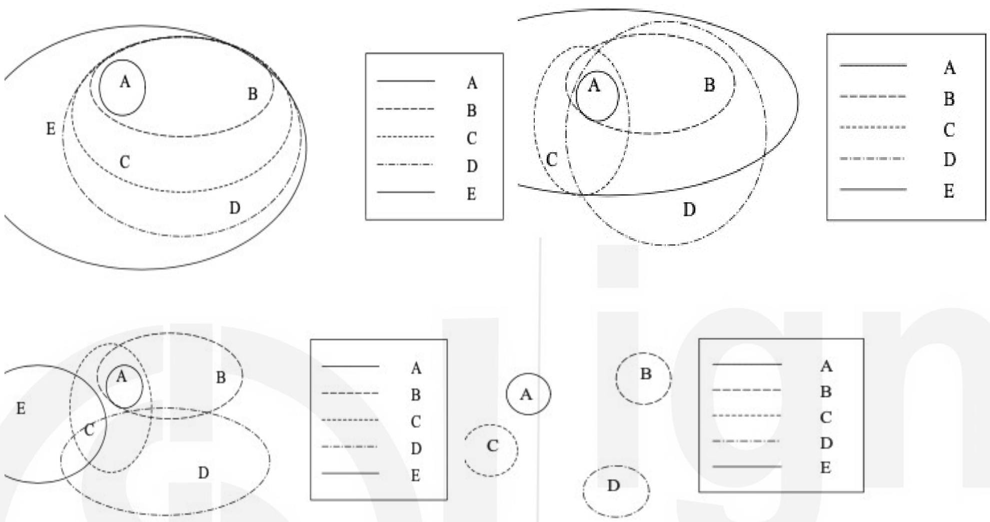


Figure 2. (a-d)

The source of information can be an entity or person giving some relevant state information. Here the information source is a non biased source of information. The information received from such sources is combined to provide more reliable information for further use. The D-S theory models are able to handle the varying precision regarding the information and hence no additional assumptions are needed to represent the information.

7.9.2 Frame of Discernment

Let us consider a random variable 'θ' whose true value is not known. Let 'θ' = { θ_1, θ_2 ….. θ_n } represent mutually exclusive and discretized values of the possible outcome of 'θ'. Conventionally, the uncertainty about 'θ' is given by the assigning probability p_i to the elements θ_i, i = 1: n, satisfying sum p_i = 1. In the case of D-S theory, the probabilities are assigned to the subsets of 'θ' and the individual element 'θ_i' along with it.

7.9.3 The Power Set P(θ= 2^{θ})

This is defined as the set of all subsets of 'θ' including singletons, defining the frame of 'θ'. The subset of this powerset may contain a single or conjunctions of hypotheses. Here, with respect to the power set, the complete probability assignment is called basic probability assignment.

The core functions in D-S theory are :

1. Basic Probability Assignment function

This is represented by m and maps the power set to the interval 0 and 1. Here, the basic probability assignment (bpa) to the null set is 0 and for all subsets of the power set sum = 1. For a given set A, m(A) represents the measure of belief assigned by the available evidences in support of A, where $A \in 2^\{\theta\}$. Mathematically, the bpa can be represented as follows.

1. m : $2^\{\theta\}$--> [0, 1] (interval)
2. $m(\phi) = 0$ (null)
3. $m(A) \geq 0, \forall A \in 2^\{\theta\}$
4. sum{m(A) $\forall A \in 2^\{\theta\}$} = 1.

This is to note here that, the element of power set with m(A) > 0 is termed as focal element(s).

Example9: Let θ = {a, b, c}; then the power set is P(θ) = {ϕ, a,b,c, (a,b), (a,c), (b,c),

(a,b,c)}. The information source assigned the m-values as m(a) = 0.2, m(c) = 0.1 and m(a,b) = 0.4. Here the mentioned three subsets are focal elements.

2. The Belief Function

The assignment of the basic probability we can define the lower and upper bounds of the intervals representing the precise probability of a set. This is also bounded by continuous measures of nonadditive nature known as Belief and Plausibility.

The lower bound (belief) for set A is defined as the sum of all basic probability assignments of proper subset B of set A. The measurement of the amount of support by the information source given to support a specific element as a correct one is done by the belief function, mathematically $Bel(A) = sum_{\{B \subset A\}}\ m(B)\ \ \forall\ A \subset \theta$.

3. The Plausibility Function

The upper bound (plausibility) for set A is defined as the sum of all basic probability assignments of B intersecting set A, mathematically $Pl(A) = sum_{\{B \cap A \neq \phi\}}\ m(B)$. Here, the plausibility function measures the level of information by a source contradicting an element as a correct answer specifically.

Apart from the above-mentioned functions a few terms also require some attention while referring

to the D-S theory. The Uncertainty Interval, shows the range where the true probability may be

found. This is calculated as the difference of belief and plausibility level i.e. Pl(A) - Bel(A).

7.9.4 Rule of Combination

In the D-S theory, the measure of Plausibility and Belief are taken from the combined assignments. The D-S rule of combination takes multiple belief functions and combines them using m i.e. respective basic probability assignments. The D-S combination rule is basically a conjunctive operation i.e. AND. Here the joint $m_{\{12\}}$ (combination) is obtained using aggregating two basic probability assignments m_1 and m_2 as following:

$$m_{\{12\}}\ (A) = 1/sum_{\{\{B \cap A = A\}\ \ m1\ (B)\ m2\ (C)\}}\{1\text{-}K\}\ ,$$

Where, $A \neq \phi$,

$M_{\{12\}}(\phi) = 0$,

And $K = \text{sum}_{\{B \cap C = \phi\}}\ m_1(B)\ m_2(C)$.

Here, in the above expression, K is the basic probability mass which is associated with the conflict calculated as a sum of products of the basic probability assignments of all sets having null intersection. The normalization factor is represented as 1-K in the denomination. The rule is associative, commutative but not continuous or idempotent in nature.

Example 10: In a multinational company 100 applicants appeared for a job interview. The company setup two interview boards for applicants.

While assessing the grades of the class of 100 students, two of the class teachers responded the overall result as follow. First teacher assessed that 40 students will get A and 20 students will get B grade amongst the total 60 students he interviewed. Whereas second teacher stated that 30 students will get A grade and 30 students will get either A or B amongst the 60 students he took the interview. Combining both evidences to find the resultant evidence, we will do following calculations. Here frame of discernment θ= {A, B} and Power set 2^{θ}= {∅, A, B, (A, B)},

Evidence (1) =Ev1	Evidence (2) =Ev2
$m_1(A) = 0.4$	$m_2(A) = 0.3$
$m_1(B) = 0.2$	$m_2(A, B) = 0.3$
$m_1(\theta) = 0.4$	$m_2(\theta) = 0.4$

Plausibility function (PI):

$A \cap A = A \neq \emptyset$ hence $m_1(A) = 0.4$

$A \cap B = \emptyset$

$A \cap \theta = A \neq \emptyset$ hence $m_1(\theta) = 0.4$

$Pl_1(A) = m_1(A) + m_1(\theta) = 0.4 + 0.4 = 0.8$

$A \cap A = A \neq \emptyset$ hence $m_2(A) = 0.3$

$A \cap B = \emptyset$

$A \cap \theta = A \neq \emptyset$ hence $m_2(\theta) = 0.4$

$Pl_2(A) = m_2(A) + m_2(\theta) = 0.3 + 0.4 = 0.7$

$B \cap A = \emptyset$

$B \cap B = B \neq \emptyset$ hence $m_1(B) = 0.2$

$B \cap \theta = B \neq \emptyset$ hence $m_1(\theta) = 0.4$

$Pl_1(B) = m_1(B) + m_1(\theta) = 0.2 + 0.4 = 0.6$

$(A, B) \cap A = A \neq \emptyset$ $m_2(A) = 0.3$

$(A, B) \cap B = B \neq \emptyset$, $m_2 (B) = 0$

$(A, B) \cap (A, B) = (A, B) \neq \emptyset$ $m_2(A, B) = 0.3$

$(A, B) \cap \theta = (A, B) \neq \emptyset$ hence $m_2 (\theta) = 0.4$

$Pl_1(A, B) = m_2 (A) + m_2 (A, B) + m_2 (\theta) = 0.3 + 0.3 + +0.4 = 1.0$

$\theta \cap A = A \neq$ □ hence $m_1 (A) = 0.4$

$\theta \cap B = B \neq$ □ hence $m_1 (B) = 0.2$

$\theta \cap \theta = \theta \neq$ □ hence $m_1 (\theta) = 0.4$

$Pl_1(\theta) = m_1 (A) + m_1 (B) + m_1 (\theta) = 0.4 + 0.2 + 0.4 = 1.0$

$\theta \cap A = A \neq$ □ hence $m_2 (A) = 0.3$

$\theta \cap (A, B) = (A, B) \neq$ □, $m_2 (A, B) = 0.3$

$\theta \cap \theta = \theta \neq$ □ hence $m_2 (\theta) = 0.4$

$Pl_2(\theta) = m_2 (A) + m_2 (A, B) + m_2 (\theta) = 0.3 + 0.3 + 0.4 = 1.0$

D-S Rule of Combination: Table 3 shows combination of concordant evidences using D-S theory.

Evidences	m1(A)=0.4		m1(B)=0.2		m1(θ)=0.4
m2(A)=0.3	m1-2 (A)	0.12	m1-2 (□)	0.06	m1-2 (A) 0.12
m2(A,B)=0.3	m1-2 (A)	0.12	m1-2 (B)	0.06	m1-2 (A,B) 0.12

m2(θ)=0.4	m1-2 (A)	0.16	m1-2 (B)	0.08	m1-2 (θ)	0.16

k = 0.06 and 1 − k = 0.94 Combined masses are worked out

Bel1-2(A) = m1-2(A) = 0.553

Bel1-2(B) = m1-2(B) = 0.149

$Bel_{1-2}(A, B) = m_{1-2}(A) + m_{1-2}(B) + m_{1-2}(A, B) = 0.553 + 0.149 + 0.128 = 0.83$ $Bel_{1-2}(\theta) = m_{1-2}(A) + m_{1-2}(B) + m_{1-2}(A, B) + m_{1-2}(\theta) = 0.553 + 0.149 + 0.128 +$

$0.170 = 1$

$Pl_{1-2}(A) = m_{1-2}(A) + m_{1-2}(A, B) + m_{1-2}(\theta) = 0.553 + 0.128 + 0.170 = 0.851,$

(85 students in A Grade)

$Pl_{1-2}(B) = m_{1-2}(B) + m_{1-2}(A, B) + m_{1-2}(\theta) = 0.149 + 0.128 + 0.170 = 0.447,$

(45 students in B Grade)

$Pl_{1-2}(A, B) = m_{1-2}(A) + m_{1-2}(B) + m_{1-2}(AB) + m_{1-2}(\theta) = 0.553 + 0.149 + 0.128 + 0.170 = 1.0$

$Pl_{1-2}(\theta) = m_{1-2}(A) + m_{1-2}(B) + m_{1-2}(A, B) += 0.553 + 0.149 + 0.128 + 0.170 =$

1.00. (100 students in total)

According to rule of combination, concluded ranges are then 55 to 85 students will get

``A'' grade and 15 to 45 students will get ``B'' grade.

Key advantages of D-S theory:

- The level of uncertainty reduces with addition of information.
- Addition of more evidences reduces ignorance
- We can represent diagnose hierarchies using D-S theory.

Check Your Progress 3

Problem-1. Differentiate between Join, Marginal and conditional probability with an example of each.

Problem-2. Explain Dempster Shafer theory with a suitable example.

Problem-3. What are different type of evidences? Give suitable example of each.

7.10 SUMMARY

This unit relates to the discussion over Reasoning with uncertain information, whih involves Review of Probability Theory, and Introduction to Bayesian Theory. Unit also covers the concept of Baye's Networks, which is later used for the purpose of inferencing. Finally, the unit discussed about the Other Paradigm of Uncertain Reasoning, including the Dempster Scheffer Theory

7.11 SOLUTIONS/ANSWERS

Check Your Progress- 1

Problem -1. In each of the following exercises, an experiment is described. Specify the relevant sample spaces:

a) A machine manufactures a certain item. An item produced by the machine is tested to determine whether or not it is defective.
b) An urn contains six balls, which are colored differently. A ball is drawn from the urn and its color is noted.
c) An urn contains ten cards numbered 1 through 10. A card is drawn, its number noted and the card is replaced. Another card is drawn and its number is noted.

Solution - *Please refer to section 7.3 to answer these problems.

Problem 2. Suppose a six-faced die is thrown twice. Describe each of the following events:

i) The maximum score is 6.
ii) The total score is 9.
iii) Each throw results in an even score.
iv) Each throw results in an even score larger than 2.
v) The scores on the two throws differ by at least 2.

Solution - *Please refer to section 7.3 to answer these problems.

Check Your Progress 2

Problem-1: In a class,three students tossed one coins (one each) for 3 times. Write down all the possible outcomes which can be obtained in this experiment.

Solution - *Please refer to example 4 and section 7.3 to solve these problems

Problem-2: In problem 1, what is the probability of getting 2 more than 2 heads at a time. Also write the probability of getting three tails at a time.

Solution - *Please refer to example 4 and section 7.3 to solve these problems

Problem-3: In problem 1 calculate the Relative frequency of tail $r_n(T)$.

Solution - *Please refer to example 4 and section 7.3 to solve these problems

Check Your Progress 3

Problem-1. Differentiate between Join, Marginal and conditional probability with an example of each.

Solution - *Please refer to section 7.9 and example 10 to answer these problems.

Problem-2. Explain Dempster Shafer theory with a suitable example.

Solution - *Please refer to section 7.9 and example 10 to answer these problems.

Problem-3. What are different type of evidences? Give suitable example of each.

Solution - *Please refer to section 7.9 and example 10 to answer these problems.

7.12 FURTHER READINGS

1. David Barber ,"Bayesian Reasoning And Machine Learning", Cambridge University Press
2. John J. Craig, "Introduction to Robotics", Addison Wesley publication
3. Ela Kumar, " Artificial Intelligence", IK International Publications
4. Ela Kumar, " Knowledge Engineering ", IK International Publications

UNIT 8 FUZZY AND ROUGH SETS

8.0 INTRODUCTION

In the earlier units, we discussed PL and FOPL systems for making inferences and solving problems requiring logical reasoning. However, these systems assume that the domain of the problems under consideration is complete, precise and consistent. But, in the real world, the knowledge of the problem domains is generally neither precise nor consistent and is hardly complete.

In this unit, we discuss a number of techniques and formal systems that attempt to handle some of these blemishes. To begin with we discuss the **fuzzy systems** that attempt to handle **imprecision** in knowledge bases, specially, due to use of natural language words like hot, good, tall etc.

Then, we discuss **non-monotonic systems** which deal with **indefiniteness** of knowledge in the knowledge bases. The significance of these systems lies in the fact that most of the statements in the knowledge bases are actually based on **beliefs** of the concerned persons or actors. These beliefs get revised as better evidence for some other beliefs become available, where the later beliefs may be in conflict with the earlier beliefs. In such cases, the earlier beliefs my have to be temporarily suspended or permanently excluded from further considerations.

Subsequently, we will discuss two formal systems that attempt to handle **incompleteness** of the available information. These systems are called **Default Reasoning Systems** and **Closed World Assumption Systems**. Finally, we discuss some inference rules, viz, **abductive** inference rule and **inductive** inference rule that are, though not deductive, yet are quite useful in solving problems arising out of everyday experience.

8.1 OBJECTIVES

After going through this unit, you should be able to:

- enumerate various formal methods, which deal with different types of blemishes like incompleteness, imprecision and inconsistency in a knowledge base;
- discuss, why fuzzy systems are required;
- discuss, develop and use fuzzy arithmetic tools in solving problems, the descriptions of which involve imprecision;
- discuss default reasoning as a tool for handling incompleteness of knowledge;
- discuss Closed World Assumption System, as another tool for handling incompleteness of knowledge, and
- discuss and use non-deductive inference rules like abduction and induction, as tools for solving problems from everyday experience.

8.2 FUZZY SYSTEMS

In the symbolic Logic systems like, PL and FOPL, that we have studied so far, any (closed) formula has a truth-value which must be binary, viz., *True or False*. However, in our everyday experience, we encounter problems, the descriptions of which involve some words, because of which, to statements of situations, it is not possible to assign a truth value: *True or False*. For example, consider the statement:

If the water is too hot, add normal water to make it comfortable for taking a bath.

In the above statement, for a number of words/phrases including 'too hot' 'add', 'comfortable' etc., it is not possible to tell when exactly water is too hot, when water is (at) normal (temperature), when exactly water is comfortable for taking a bath.

For example, we cannot tell the temperature T such that for water at temperature T or less, truth value False can be associated with the statement '*Water is too hot'* and at the same time truth-value True can also be associated to the same statement '*Water is too hot'* when the temperature of the water is, say, at degree T + 1, T + 2….etc.

Some other cases of Fuzziness in a Natural Language

***Healthy* Person:** we cannot even enumerate all the parameters that **determine** health. Further, it is even more difficult to tell for what value of a particular parameter, one is healthy or otherwise.

***Old/young person*:** It is not possible to tell exactly upto exactly what age, one is young and, by just addition of one day to the age, one becomes old. We age gradually. Aging is a **continuous** process.

***Sweet* Milk:** Add small sugar cube one at a time to glass of milk, and go on adding upto, say, 100 small cubes.

Initially, without sugar, we may take milk as not sweet. However, with addition of each one small sugar particle cube, the sweetness **gradually** increases. It is not possible to say that after addition of 100 small cubes of sugar, the milk becomes sweet, and, till addition of 99 small cubes, it was not sweet.

Pool, Pond, Lake,….., Sea, Ocean: for different sized water bodies, we can not say when exactly a pool becomes a pond, when exactly a pond becomes a lake and so on.

One of the reasons, for this type of problem of our inability to associate one of the two-truth values to statements describing everyday situations, is due to the use of natural language words like hot, good, beautiful etc. Each of these words does not denote something constant, but is a sort of linguistic variable. The context of a particular *usage* of such a word may delimit the scope of the word as a linguistic variable. The range of values, in some cases, for some phrases or words, may be very large as can be seen through the following three statements:

- Dinosaurs ruled the earth for a long period (*about millions of years*)
- It has not rained *for a long period (say about six months).*
- I had to wait for the doctor *for a long period (about six hours)*.

Fuzzy theory provides means to handle such situations. A **Fuzzy theory** may be thought as a technique of providing '**continuization**' to the otherwise binary disciplines like Set Theory, PL and FOPL.

Further, we explain how using fuzzy concepts and rules, in situation like the ones quoted below, we, the human beings solve problems, despite ambiguity in language.

Let us recall the case of crossing a road discussed in Unit 1 of Block 1. We

Mentioned that a step by step method of crossing a road may consist of

(i) Knowing (exactly) the distances of various vehicles from the path to be followed to cross over.

(ii) Knowing the velocities and accelerations of the various vehicles moving on the road within a distance of, say, one kilometer.

(iii) Using Newton's Laws of motion and their derivatives like $s = ut + \frac{1}{2}at^2$, and calculating the time that would be taken by each of the various vehicles to reach the path intended to be followed to cross over.

(iv) Adjusting dynamically our speeds on the path so that no collision takes place with any of the vehicle moving on the road.

But, we know the human beings not only do not follow the above precise method but cannot follow the above precise method. **We, the human beings rather feel comfortable with fuzziness than precision.** We feel comfortable, if the instruction for crossing a road is given as follows:

Look on both your *left hand* and *right hand* sides, particularly in the beginning, to your right hand side. If there is no vehicle within *reasonable* distance, then attempt to cross the road. You may have to retreat back while crossing, from *somewhere* on the road. Then, try again.

The above instruction has a number of words like *left*, *right* (it may 45° to the right or 90° to the right) *reasonable*, each of which does not have a definite meaning. But we feel more comfortable than the earlier instruction involving precise terms.

Let us consider another example of *our being comfortable with imprecision than precision*. The statement: '*The sky is densely clouded*' is more comprehensible to human beings than the statement: '*The cloud cover of the sky is 93.5 %*'.

Thus is because of the fact that, we, the human beings are still better than computers in **qualitative** reasoning. Because of better qualitative reasoning capabilities

- just by looking at the eyes only and/or nose only, we may recognize a person.
- just by taking and feeling a small number of grains from cooking rice bowl, we can **tell** whether the rice is properly cooked or not.
- just by looking at few buildings, we can identify a locality or a city.

Achieving Human Capability

In order that computers achieve human capability in solving such problems, computers must be able to solve problems for which ***only incomplete and/or imprecise*** information/knowledge is available.

Modelling of Solutions and Data/Information/Knowledge

We know that for any problem, the plan of the proposed solution and the relevant information is fed in the computer in a form acceptable to the computer.

However, the problems to be solved with the help of computers are, in the first place, felt by the human beings. And then, the plan of the solution is also prepared by human beings.

It is conveyed to the computer mainly for execution, because computers have much better executional speed.

Summarizing the discussion, we conclude the following facts

(i) We, the human beings, sense problems, desire the problems to be solved and express the problems and the plan of a solution using imprecise words of a natural language.

(ii) We use computers to solve the problems, because of their executional power.

(iii) Computers function better, when the information is given to the computer in terms of ***mathematical entities*** like numbers, sets, relations, functions, vectors, matrices graphs, arrays, trees, records, etc., and when the ***steps of solution*** are generally precise, involving no ambiguity.

In order to meet the mutually conflicting requirements:

(i) *Imprecision* of natural language, with which the human beings are comfortable, where human beings feel a problem and plan its solution.

(ii) *Precision* of a formal system, with which computers operate efficiently, where computers execute the solution, generally planned by human beings

a new formal system viz. Fuzzy system based on the concept of 'Fuzzy' was suggested for the first time in 1965 by L. Zadeh.

In order to initiate the study of Fuzzy systems, we quote two statements to recall the difference between a precise statement and an imprecise statement.

A precise Statement is of the form: 'If income is more than 2.5 lakhs then tax is 10% of the taxable income'.

An ***imprecise*** statement may be of the form: 'If the ***forecast*** about the rain being **slightly less** than previous year ***is believed***, then there is around 30% **probability** that economy may suffer heavily'.

The **concept of 'Fuzzy',** which when applied as a **prefix/adjective to mathematical entities like set, relation**, **functions, tree,** etc., **helps us in modelling** the **imprecise** data, information or knowledge **through mathematical tools**.

Crisp Set/Relation vs. Fuzzy Set/Relation: In order to differentiate the sets, normally used so far, from the *fuzzy sets* to be introduced soon, we may call the normally called sets as *crisp sets*.

Next, we explain, how the fuzzy sets are defined, using mathematical entities, **to capture imprecise concepts**, through an example of the concept : tall.

In Indian context, we may say, a ***male adult***, is

(i) **definitely tall** if his height > 6 feet

(ii) **not at all** tall if height < 5 feet and

(iii) if his height = 5′ *2”* a **little bit tall**

(iv) if his height = 5′ *6”* **slightly tall**

(v) if height = 5′ *9”* **reasonably tall** etc.

Next step is **to model 'definitely tall' 'not at all tall', 'little bit tall', 'slightly tall' 'reasonably Tall'** etc. in terms of mathematical entities, e.g., numbers; sets etc.

In **modelling the vague concept like 'tall', through fuzzy sets,** the numbers in the **closed set [0, 1] of reals** may be used on the following lines:

(i) '***Definitely tall'*** may be represented as '***tallness having value 1***'

(ii) '***Not at all tall***' may be represented as '***Tallness having value 0'***

other adjectives/adverbs may have values between 0 and 1 as follows:

(iii) '***A little bit tall***' may be represented as '***tallness having value say .2'***.

(iv) '***Slightly tall***' may be represented as '***tallness having value say .4'***.

(v) '***Reasonably tall***' may be represented as '***tallness having value say .7'***.

and so on.

Similarly, the values of other concepts or, rather, other **linguistic variables like sweet, good, beautiful**, etc. may be considered **in terms of real numbers between 0 and 1.**

Coming back to the ***imprecise concept of tall***, let us think of five male persons of an organisation, viz., Mohan, Sohan, John, Abdul, Abrahm, with heights 5' 2", 6' 4",

5' 9", 4' *8", 5' 6"* respectively.

Then had we talked only of crisp set of tall persons, we would have denoted the

Set of tall persons in the organisation = {Sohan}

But, a fuzzy set, representing tall persons, include ***all the persons alongwith respective degrees of tallness***. Thus, ***in terms of fuzzy sets,*** we write:

Tall = {Mohan/.2; Sohan/1; John/.7; Abdul/0; Abrahm/.4}.
The values .2, 1, .7, 0, .4 are called ***membership values or degrees***:

Note: ***Those elements which have value 0 may be dropped*** e.g.

Tall may also be written as Tall = {Mohan/.2; Sohan/1; John/.7;, Abrahm/.4}, neglecting Abdul, with associated degree zero.

If we ***define short/Diminutive*** as exactly ***opposite of Tall*** we may say

Short = {Mohan/.8; Sohan/0; John/.3; Abdul/1; Abrahm/.6}

8.3 INTRODUCTION TO FUZZY SETS

In the case of ***Crisp sets***, we have the concepts of *Equality of sets, Subset of a set, and Member of a set*, as illustrated by the following examples:

(i) **Equality of two sets**

Let A = {1, 4, 3, 5}

B = {4, 1, 3, 5}

C = {1, 4, 2, 5}

be three given sets.

Then, Set A is equal to set B *denoted by A = B*. But A is not equal to C, *denoted by*

$A \neq C$.

(ii) **Subset**

Consider sets A = {1, 2, 3, 4, 5, 6, 7}

B = {4, 1, 3, 5}

C = {4, 8}

Then B is a subset of A, denoted by $B \subset A$. Also C is not a subset of A, denoted by

$C \not\subset A$.

(iii) **Belongs to/is a member of**

If A = {1, 4, 3, 5}

Then each of 1, 4, 3 and 5 is called an *element or member* of A and the fact that *1 is a member of A* is denoted by $1 \in A$.

Corresponding Definitions/ concepts for Fuzzy Sets

In order to define for fuzzy sets, the concepts corresponding to the concepts of *Equality of Sets*, *Subset* and *Membership of a Set* considered so far only for crisp sets, first we illustrate the concepts through an example:

Let X be the set on which fuzzy sets are to be defined, e.g.,

X = {Mohan, Sohan, John, Abdul, Abrahm}.

Then X is called the ***Universal Set.***

Note: *In every fuzzy set, all the elements of X with their corresponding memberships values from 0 to 1, appear.*

(i) Degree of Membership: In respect of fuzzy sets, we do not speak of just '*membership*', but speak of '*degree of membership*'.

In the set

A = {Mohan/.2; Sohan/1; John/.7; Abrahm/.4},

Degree (Mohan) = .2, degree (John) =.4

For (ii) Equality of Fuzzy sets: Let A, B and C be fuzzy sets defined on X as follows:

Let A = {Mohan/.2; Sohan/1; John/.7; Abrahm/.4}

B = {Abrahm/.4, Mohan/.2; Sohan/1; John/.7}.

Then, as degrees of each element in the two sets, equal; we say fuzzy set A equals fuzzy set B, denoted as A = B

However, if C = {Abrahm/.2, Mohan/.4; Sohan/1; John/.7}, then

$A \neq C$.

(iii) Subset/Superset

Intuitively, we know

(i) The **Set of 'Very Tall' people** should be a **subset** of the set of **Tall people**.

(ii) If the **degree of 'tallness'** of a person **is say .5** then degree of **'Very Tallness'** for the person should be **lesser say .3**.

Combining the above two ideas we, may say that if

A = {Mohan/.2; Sohan/1; John/.7; Abrahm/.4} and

B = {Mohan/.2, Sohan/.9, John/.6, Abraham/.4}and further,

C = {Mohan/.3, Sohan/.9, John/.5, Abraham/.4},

then, in view of the fact that for each element, degree in A is greater than or equal to degree in B, **B is a subset of A** denoted as $B \subset A$.

However, degree (Mohan) = .3 in C and degree (Mohan) =.2 in A,

,therefore, C is **not** a subset of A.

On the other hand degree (John) = .5 in C and degree (John) = .7 in A,

therefore, A is also not a subset of C.

We generalize the ideas illustrated through examples above

Let ***A and B be fuzzy sets*** on the universal set $X = \{x_1, x_2, \ldots, x_n\}$ ***(X is called the Universe or Universal set)*** such that

$A = \{x_1/v_1, x_2/v_2, \ldots, x_n/v_n\}$ and $B = \{x_1/w_1, x_2/w_2, \ldots, x_n/w_n\}$

with that $0 \le v_i$, $w_i \le 1$. Then fuzzy set A equals fuzzy set B, denoted as A = B, *if and only if* $v_i = w_i$ for all i = 1,2,….,n. Further if $w \le v_i$ ***for all i. then B is a fuzzy subset of A.***

Example: Let X = {Mohan, Sohan, John, Abdul, Abrahm}

A = {Mohan/.2; Sohan/1; John/.7; Abrahm/.4}

B = {Mohan/.2, Sohan/.9, John/.6, Abraham/.4}

Then B is a fuzzy subset of A.

In respect of fuzzy sets vis-à-vis (crisp) sets, we may note that:

- Corresponding to the concept of '**belongs to**' of **(Crisp) set**, we use the concept of '**degree of membership**' for fuzzy sets.
- It may be noted that every **crisp set** may be thought of as a **Fuzzy Set,** but **not conversely**. For example, if **Universal set is**

X = {Mohan, Sohan, John, Abdul, Abrahm} and

A = set of those members of X who are ***at least graduates,*** say,

= {Mohan, John, Abdul}

then we **can rewrite A as a fuzzy set** as follows:

A = {Mohan/1; **Sohan**/0; John/1; Abdul/1; **Abrahm**/0}, in which degree of each member of the crisp set, is taken as one and degree of each element of the universal set which does not appear in the set A, is taken as zero.

However, conversely, a fuzzy set **may not** be written as a **crisp set.** Let C be a fuzzy set denoting **Educated People,** where **degree of education is defined as** follows:

degree of education (Ph.D. holders) = 1

degree of education (Masters degree holders) = 0.85

degree of education (Bachelors degree holders) = .6

degree of education (10 + 2 level) = 0.4

degree of education (8^{th} Standard) = 0.1

degree of education (less than 8th) = 0.

Let us C = {Mohan/.85; Sohan/.4; John/.6; Abdul/1; Abrahm/0}.

Then, we cannot think of C as a crisp set.

Next, we define some more concepts in respect of fuzzy sets.

Definition: Support set of a Fuzzy Set, say C, is a crisp set, say D, containing all the elements of the universe X for which **degree** of membership in Fuzzy set **is positive. Let us consider again**

C = {Mohan/.85; Sohan/.4; John/.6; Abdul/1; Abrahm/0}.

Support of C = D = {Mohan, Sohan, John, Abdul}, where **the element Abrahm does not belong to D, because, it has degree 0 in C.**

Definition: Fuzzy Singleton is a fuzzy set in which there is exactly one element which has positive membership value.

Example:

Let us define a fuzzy set OLD on universal set X in which degree of OLD is zero if a person in X is below 20 years and Degree of Old is .2 if a person is between 20 and 25 years and further suppose that

Old = C = {Mohan/0; Sohan/0; John/.2; Abdul/0; Abrahm/0},

then support of old = **{John}** and hence old is a fuzzy singleton.

Check Your Progress - 1

Ex. 1: Discuss equality and subset relationship for the following fuzzy sets defined on the Universal set X = { a, b , c, d, e}

A = { a/.3, b/.6, c/.4 d/0, e/.7}

B = {a/.4, b/.8, c/.9, d/.4, e/.7}

C = {a/.3, b/.7, c/.3, d/.2, e/.6}

8.4 FUZZY SET REPRESENTATION

For Crisp sets, we have the operations of **Union, intersection & complementation**, as illustrated by the example:

Let X = $\{x_1, x_2, \ldots, x_{10}\}$

A = $\{x_2, x_3, x_4, x_5\}$

$B = \{x_1, x_3, x_5, x_7, x_9\}$

Then $A \cup B = \{x_1, x_2, x_3, x_4, x_5, x_7, x_9\}$

$A \cap B = \{x_3, x_5\}$

A' or $X \sim A = \{x_1, x_6, x_7, x_8, x_9, x_{10}\}$

The concepts of Union, intersection and complementation for **crisp sets may be extended to FUZZY** sets after observing that for crisp sets A and B, we have

(i) $A \cup B$ is the **smallest** subset of X **containing** both A and B.

(ii) $A \cap B$ is the **largest** subset of X **contained in** both A and B.

(iii) **The complement A' is such that**

(a) A and A' **do not have any element in common** and

(b) Every element of the universal set **is in either A or A'.**

Fuzzy Union, Intersection, Complementation:

In order to motivate proper definitions of these operations, we may recall

(1) when a **crisp set** is treated as a **fuzzy set** then

(i) membership in a crisp set is indicated by degree/value of membership as 1 (one) in the corresponding Fuzzy set,

(ii) non-membership of a crisp set is indicated by degree/value of membership as **zero** in the corresponding Fuzzy Set.

Thus, **smaller the value** of degree of membership, a sort of **lesser it is a member** of the Fuzzy set.

(2) While taking union of Crisp sets, members of both sets are included, and none else. However, in each Fuzzy set, all members of the universal set occur but their degrees determine the level of membership in the fuzzy set.

The facts under (1) and (2) above, lead us to define:

The **Union of two fuzzy sets** A and B, is the set C with the same universe as that of A and B such that, the degree of an element of C is equal to the **MAXIMUM** of degrees of the element, in the two fuzzy sets.

(*if Universe A $\neq$ Universe B, then take Universe C as the union of the universe A and universe B)*

The **Intersection C of two fuzzy sets A and B is the fuzzy set in which**, the degree of an element of C is equal to the **MINIMUM** of degrees in the two fuzzy sets.

Example:

A = {Mohan/.85; Sohan/.4; John/.6; Abdul/1; Abrahm/0}

B = {Mohan/.75; Sohan/.6; John/0; Abdul/.8; Abrahm/.3}

Then

$A \cup B$ = {Mohan/.85; Sohan/.6; John/.6; Abdul/1; Abrahm /.3}

$A \cap B$ = {Mohan/.75; Sohan/.4; John/0; Abdul/.8; Abrahm/0} and, the complement of A denoted by A′ is given by

C′ = {Mohan/.15; Sohan/.6; John/.4; Abdul/0; Abrahm /1}

Properties of Union, Intersection and Complement of Fuzzy Sets:

The following properties which hold for ordinary sets, also, hold for fuzzy sets

Commutativity

(i) $A \cup B = B \cup A$

(ii) $A \cap B = B \cap A$

We prove only (i) above just to explain, how the involved equalities, may be proved in general.

Let $U = \{x_1, x_2 \ldots\ldots x_n\}$. *be universe for fuzzy sets A and B*

If $y \in A \cup B$, *then y is of the form {xi/di} for some i*

$y \in A \cup B \Rightarrow y = \{xi/ei\}$ *as member of A and*

$y = (xi/fi\}$ *as member of B and*

$di = \max \{ei, fi\} = \max \{fi, ei\}$

$\Rightarrow y \in B \cup A.$

Rest of the properties are stated without proof.

Associativity

(i) $(A \cup B) \cup C = A \cup (B \cup C)$

(ii) $(A \cap B) \cap C = A \cap (B \cap C)$

Distributivity

(i) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

(ii) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

DeMorgan's Laws

$(A \cup B)' = A' \cap B'$

$(A \cap B)' = A' \cup B'$

Involution or Double Complement

$(A')' = A$

Idempotence

$A \cap A = A$

$A \cup A = A$

Identity

$A \cup U = U \quad A \cup U = A$

$A \cap \phi = A \quad \phi \cap A = \phi$

where ϕ: empty fuzzy set = {x/0 with x$\in U$} and U: universe = {x/1 with x$\in U$}

Check Your Progress - 2

Ex. 2: For the following fuzzy sets

A = {a/.5, b/.6, c/.3, d/0, e/.9} and

B = { a/.3, b/.7, c/.6, d/.3, e/.6},

find the fuzzy sets A $\cap$ B, A $\cup$ B and (A $\cap$ B)'

Next, we discuss three operations, viz., *concentration, dilation and normalization*, that are relevant only to fuzzy sets and can not be discussed for (crisp) sets.

(1) Concentration of a set A is defined as

CON (A) = $\{x/m^2_A(x)|x\in U\}$

Example:

If A = {Mohan/.5; Sohan/.9; John/.7; Abdul/0; Abrahm/.2}

then

CON (A) = {Mohan/.25; Sohan/.81; John/.49; Abdul/0; Abrahm/.04}.

In respect of concentration, it may be noted that the associated values being between 0 and 1, on squaring, become smaller. In other words, the values concentrate towards zero. This fact may be used for giving increased emphasis on a concept. If *Brightness* of articles is being discussed, then *Very bright* may be obtained in terms of

CON. (Bright).

(2) Dilation (Opposite of Concentration) of a fuzzy set A is defined as

DIL (A) = $\{x/\sqrt{m_A(x)}|x \in U\}$

Example:

If A = {Mohan/.5; Sohan/.9; John/.7; Abdul/0; Abrahm/.2}

then

DIL (A) = {Mohan/.7; Sohan/.95; John/.84; Abdul/0; Abrahm/.45}

The associated values, that are between 0 and 1, on taking square-root get increased, e.g., if the value associated with x was .01 before dilation, then the value associated with x after dilation becomes .1, i.e., ten times of the original value.

This fact may be used for *decreased emphasis*. For example, if colour say '*yellow*' has been considered already, then '*light yellow*' may be considered in terms of already discussed '*yellow*' through Dilation.

(3) Normalization of a fuzzy set, **is defined as** $NORM(A) = \left\{x/\left(\frac{m_A(x)}{Max}\right)|\, x \in U\right\}$.

NORM (A) and is a fuzzy set in which membership values are obtained by dividing values of the membership function of A **by the maximum membership function**.

The resulting fuzzy set, called the **normal**, (or **normalized) fuzzy set**, has the maximum of membership function value of 1.

Example:

If A = {Mohan/.5; Sohan/.9; John/.7; Abdul/0; Abrahm/.2}

Norm (A) = {Mohan/(.5 $\div$.9 = .55.); Sohan/1; John /(.7 $\div$.9 = .77.); Abdul/0; Abrahm/(.2 $\div$.9 = .22.)}

Note: If one of the members has value 1, then Norm (A) = A,

Relation & Fuzzy Relation

We know from our earlier background in Mathematics that a relation from a set A to a set B is a subset of A x B.
For example, The relation of father may be written as {{Dasrath, Ram), …}, *which is a subset of A × B, where A and B are sets of persons living or dead.*

The relation of *Age* may be written as

{(Mohan, 43.7), (Sohan, 25.6), …},

where A is set of living persons and B is set of numbers denoting years.

Fuzzy Relation
In fuzzy sets, every element of the universal set occurs with some degree of membership. **A fuzzy relation may be defined in different ways.** One way of defining fuzzy relation is to assume the underlying sets as crisp sets. We will discuss only this case.

Thus, a **relation from A to B, where we assume A and B as crisp sets, is a fuzzy set, in which with each** element of A × B is associated a degree of membership between zero and one.

For example:

We may define the relation of UNCLE as follows:

(i) *x* is an UNCLE of *y* with degree **1** if *x* is brother of mother or father,

(ii) *x* is an UNCLE of *y* with degree **.7** if *x* is a brother of an UNCLE of *y*, and *x* is not covered above,

(iii) *x* is an UNCLE of *y* with degree **.6** if *x* is the son of an UNCLE of mother or father.

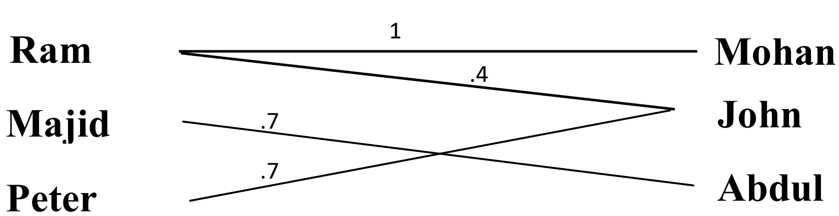
Now suppose

Ram is UNCLE of Mohan with degree **1,** Majid is UNCLE of Abdul with degree **.7** and Peter is UNCLE of John with degree **.7**. Ram is UNCLE of John with degree**.4**

Then **the relation of UNCLE can be written** as a set of ordered-triples as follows:

{(Ram, Mohan, 1), (Majid, Abdul, .7), (Peter, John, .7), (Ram, John, .4)}.

As in the case of ordinary relations, we can use matrices and graphs to represent FUZZY relations, e.g., the relation of UNCLE discussed above, may be graphically denoted as



Fuzzy Graph

Fuzzy Reasoning
In the rest of this section, we just have a fleeting glance on Fuzzy Reasoning.

Let us recall the well-known Crisp Reasoning Operators

(i) AND

(ii) OR

(iii) NOT

(iv) *IF* P *THEN* Q

(v) P *IF AND ONLY IF* Q

Corresponding to each of these operators, there is a fuzzy operator discussed and defined below. For this purpose, we assume that P and Q are fuzzy propositions with associated degrees, respectively, deg (P) and deg (Q) between 0 and 1.

The deg (P) = 0 denotes *P is False* and deg (P) =1 denotes *P is True.*

Then the operators are defined as follows:

(i) Fuzzy AND to be denoted by $\wedge$, is defined as follows:

For given fuzzy propositions P and Q, the expression P $\wedge$ Q denotes a fuzzy proposition with Deg (P $\wedge$ Q) = **min** (deg (P), deg (Q))

Example: Let P: *Mohan is tall* with deg (P) = .7

Q: *Mohan is educated* with deg (Q) = .4

Then P $\wedge$ Q denotes: *'Mohan is tall and educated'* with degree ((**min**) {.7, .4}) = **.4**

(ii) Fuzzy OR to be denoted by $\vee$, is defined as follows:

For given fuzzy propositions P and Q, P $\vee$ Q is a fuzzy proposition with

Deg (P $\vee$ Q) = **max** (deg (P), deg (Q))

Example: Let P: *Mohan is tall* with deg (P) = .7

Q*: Mohan is educated* with deg (Q) = .4

Then P $\vee$ Q denotes*: 'Mohan is tall or educated'* with degree ((**max**) {.7, .4}) = **.7**

8.5 FUZZY REASONING

The Fuzzy Reasoning is taken care by the following systems in general:

1) Non Monotonic reasoning Systems
2) Default Reasoning Systems
3) Closed World Assumption Systems

Let's start our discussion with the understanding of Non Monotonic Reasoning Systems

1) NON-MONOTONIC REASONING SYSTEMS

Monotonic Reasoning: The conclusion drawn in PL and FOPL are only through (valid) deductive methods. When some axiom is *added* to a PL or an FOPL system, then, through deduction, we can draw *more* conclusions. Hence, more additional facts become available in the knowledge base with the addition of each axiom. Adding of axioms to the knowledge base increases the amount of knowledge contained in the knowledge base. Therefore, the set of facts through inferences in such systems **can only grow larger** with addition of each axiomatic fact. Adding of new facts can not reduce the size of K.B. Thus, amount of knowledge **monotonically** increases with the number of independent premises due to new facts that become available.

However, in everyday life, many times in the light of new facts that become available, we may have to revise our earlier knowledge. For example, we consider a sort of deductive argument in FOPL:

(i) Every bird can fly long distances

(ii) Every pigeon is a bird. (iii) Tweety is a pigeon.

Therefore, Tweety can fly long distances.

However, later on, we come to know that Tweety is actually a hen and a hen cannot fly long distances. Therefore, we have to revise our belief that Tweety can fly over long distances.

This type of situation is not handled by any monotonic reasoning system including PL and FOPL .This is appropriately handled by Non-Monotomic Reasoning Systems, which are discussed next.

A **non-monotomic reasoning system** is one which allows *retracting* **of old knowledge due to discovery of new facts** which contradict or invalidate a part of the current knowledge base. Such systems also take care that retracting of a fact may necessitate a chain of retractions from the knowledge base or even reintroduction of earlier retracted ones from K.B. Thus, *chain-shrink* and *chain emphasis* of a K.B and reintroduction of earlier retracted ones are part of a non-monotomic reasoning system.
To meet the requirement for reasoning in the real-world, we need non-monotomic reasoning systems also, in addition to the monotomic ones. This is true specially, in view of the fact that it is not reasonable to expect that **all the knowledge needed** for a set of tasks could be acquired, validated, and loaded into the system just at the outset. In general, initial knowledge is an *incomplete* set of *partially true* facts. The set may also be redundant and may contain inconsistencies and other sources of uncertainty.

Major components of a Non-Monotomic reasoning system

Next, we discuss a typical non-monotomic reasoning system **(NMRS)** consists of the

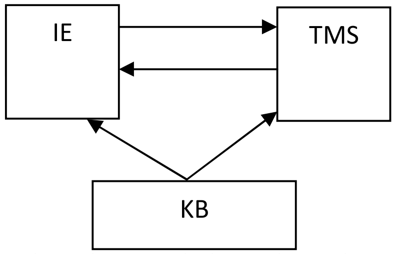
following three major components:

(1) Knowledge base (KB),

(2) Inference Engine (IE),

(3) Truth-Maintenance System (TMS).

The **KB** contains information, facts, rules, procedures etc. relevant to the type of problems that are expected to be solved by the system. The component **IE** of NMRS gets facts from KB to draw new inferences and sends the new facts discovered by it (i.e., IE) to KB. The component **TMS**, after addition of new facts to KB. either from the environment or through the user or through IE, checks for validity of the KB. It may happen that the new fact from the environment or inferred by the IE may conflict/contradict some of the facts already in the KB. In other words, an *inconsistency* may arise. In case of inconsistencies, TMS retracts some facts from KB. Also, it may *lead to a chain of retractions* which may require interactions between KB and TMS. Also, some new fact either from the environment or from IE, may invalidate some earlier retractions *requiring reintroduction* of earlier retracted facts. This may lead to a chain of reintroductions. These retrievals and introductions are taken care of by TMS. The IE is completely relieved of this responsibility. Main job of IE is *to conclude* new facts when it is supplied a set of facts.



Next, We explain the ideas discussed above through an example:

Let us assume KB has two facts P and $\sim Q \rightarrow \sim P$ and a rule called *Modus Tollens*. When **IE** is supplied these knowledge items, it concludes Q and sends Q to KB. However, through interaction with the environment, **KB** is later supplied with the information that ~ P is more appropriate than P. Then **TMS,** on the addition of ~ P to KB, finds that KB is no more consistent, at least, with P. The knowledge that ~ P is more appropriate, suggests that P be retracted. Further Q was concluded *assuming P as True*. But, in the new situation in which P is assumed to be not appropriate, Q also becomes inappropriate. **P and Q are not deleted from KB, but are just marked as dormant or ineffective.** This is done in view of the fact that later on, if again, it is found appropriate to include P or Q or both, then, instead of requiring some mechanism for adding P and Q, we just remove marks that made these dormant.

Non-monotomic Reasoning Systems deal with

1) Revisable belief systems

2) incomplete K.B. ——— Default Reasoning

Closed World assumption

2) DEFAULT REASONING

In the previous section, we discussed *uncertainty due to beliefs* (which are not necessarily *facts*) where beliefs are changeable. Here, we discuss *another form of uncertainty* that occur as a result of *incompleteness* of the available knowledge at a particular point of time.

One method of handling uncertainty due to **incomplete** KB is through **default reasoning** which is also a form of non-monotomic reasoning and is based on the following mechanism:

*Whenever, for any entity relevant to the application, information is not in the KB, then a **default value** for that type of entity, is assumed and is assigned to the entity. The default assignment is not arbitrary but is based on experiments, observations or some other rational grounds. However, the typical value for the entity is removed if some information contradictory to the assumed or default value becomes available.*

The advantage of this type of a reasoning system is that we need not store all facts regarding a situation. **Reiter has given one theory of default reasoning, which is expressed as**

$$\frac{a(x) : M\,b_1(x),.....,Mb_k(x)}{C(x)} \qquad \text{(A)}$$

where M is a *consistency operator.*

The inference rule (A) states that if a(x) is true and none of the conditions b_k (x) is in conflict or contradiction with the K.B, then you can deduce the statement C(x)

The idea of default reasoning is explained through the following example:

Suppose we have

(i) $$\frac{\text{Bird}(x) : \text{Mfly}(x)}{\text{Fly}(x)}$$

(ii) Bird (twitty)

M fly (x) stands for a statement of the form *'KB does not have any statement of the form that says x does not have wings etc, because of which x may not be able to fly'*. In other words, *Bird (x) : M fly (x)* may be taken to stand for the statement '*if x is a normal bird and if the normality of x is not contradicted by other facts and rules in the KB.*' then we can assume that *x can fly.* Combining with *Bird (Twitty),* we conclude that if KB does not have any facts and rules from which, it can be inferred that *Twitty can not fly*, then, we can conclude that *twitty can fly.*

Further, suppose, KB also contains

(i) Ostrich (twitty)

(ii) Ostrich (x) $\rightarrow$ ~ FLY (x).

From these two facts in the K.B., it is concluded that Twitty being an ostrich, can not fly. In the light of this knowledge the fact that Twitty can fly has to be withdrawn. Thus, Fly (twitty) would be locked. Because, default Mfly (Twitty) is now inconsistent.

Let us consider another example:

$$\frac{\text{Adult}(x) : \text{M drive}(x)}{\text{Drive}(x)}$$

The above can be interpreted in the default theory as:

If a person x is an adult and in the knowledge base there is no fact (*e.g., x is blind, or x has both of his/her hands cut in an accident etc*) which tells us something making x incapable of driving, **then *x*** can drive, is assumed.

3) CLOSED WORLD ASSUMPTION

Another mechanism of handling incompleteness of a KB is called 'Closed World Assumption' (CWA).

This mechanism is useful in applications where *most of the facts are known* and therefore it is reasonable *to assume that if a proposition cannot be proved, then it is FALSE. This is called CWA with failure as negation.*

This means if a ground atom P(a) is not provable, then assume ~ P(a). *A predicate like*

LESS (x, y) becomes a ***ground atom*** *when the variables x and y are replaced by constants say x by 2 and y by 3, so that we get the ground atom LESS (2, 3).*

Example of an application where CWA is reasonable is that of *Airline reservation* where city-to-city flight not explicitly entered in the flight schedule or time table, *are assumed not to exist*.

AKB is **complete** if for each ground atom P(a); either P(a) or ~ P(a) can be proved.

By the use of CWA any incomplete KB becomes complete **by the addition of the meta rule**:

If P(a) can not be proved then assume ~ P (a).

Example of an incomplete K.B: Let our KB contain only

(i) P(a).

(ii) P(b).

(iii) P(a) $\rightarrow$ Q(a).

(iv) Rule of Modus Ponens: From P and P $\rightarrow$ Q, conclude Q.

The above KB is *incomplete* as we can not say anything about Q(b) (or ~ Q(b)) from the given KB.

Remarks: In general, KB argumented by CWA need *not be* consistent i.e.,

it may contain two mutually conflicting *wffs*. For example, if our KB contains

only P(a) $\vee$ Q(b).

(Note: from P (a) $\vee$ Q (b), we can not conclude either of P (a) and Q (b) with definiteness)

As neither P(a) nor Q(b) is provable, therefore, we add ~ P(a) and ~ Q(b) by using CWA.

But, then, the set of P(a) $\vee$ Q(b), ~ P(a) and ~Q(b) is inconsistent.

8.6 FUZZY INFERENCE

PL and FOPL are *deductive* inferencing systems: i.e., the conclusions drawn are *invariably true* whenever the premises are *true*. However, due to limitations of these systems for making inferences, as discussed earlier, we must have other systems inferences. In addition to *Default Reasoning systems* and *Closed World Assumption systems*, we have the following useful reasoning systems:

1) **Abductive inference** System, which is based on the use of causal knowledge to explain and justify a (*possibly invalid*) conclusion.

   Abduction Rule **(P $\rightarrow$ Q , Q) / P**

Note that *abductive inference rule is different form Modus Ponens inference rule* in

that in abductive inference rule, the *consequent* of P $\rightarrow$ Q, i.e., Q is assumed to be

given as True and the *antecedent* of P $\rightarrow$ Q, i.e., P *is inferred.*

The abductive inference is useful in *diagnostic applications*. For example while

diagnosing a disease (*say P*), the doctor asks for the symptoms (*say Q*). Also, the doctor knows that for given the disease, say, Malaria (P); the symptoms include high fever starting with feeling of cold etc. (Q)

i.e., doctor knows P$\rightarrow$Q

The doctor then attempts to diagnose the disease (i.e., P) from symptoms. However, it should be noted that the conclusion of the disease from the symptoms may not always be correct. In general, abductive reasoning leads to correct conclusions, but the conclusions may be *incorrect* also. In other words, Abductive reasoning is not a **valid form** of reasoning.

Inductive Reasoning is a method of generalisation from a finite number of instances. The rule, generally, denoted as $\dfrac{P(a_1),\ P(a_2)......,P(a_n)}{(x)\ P(x)}$, states that from n instances $P(a_i)$ of a predicate/property *P(x),* we infer that *P(x) is True for all x.*

Thus, from a finite number of observations about some property of objects, we generalize, i.e., make a *general* statement *about all* the elements of the domain in respect of the property.

For example, we may, conclude that: *all cows are white,* after observing a large number of white cows. However, this conclusion may have some exception in the sense that we may come across a black cow also. Inductive Reasoning like Abductive Reasoning, Closed World Assumption Reasoning and Default Reasoning is not *irrefutable*. In other words, these reasoning rules lead to conclusions, which may be True, but not necessarily always.

However, all the rules discussed under Propositional Logic (PL) and FOPL, including Modus Ponens etc are deductive i.e., lead to irrefutable conclusions.

8.7 ROUGH SET THEORY

Rough set theory can be regarded as a new mathematical tool for imperfect data analysis. The theory has found applications in many domains, such as decision support, engineering, environment, banking, medicine and others. It is a mechanism to deal with imprecise/imprecise knowledge, dealing with such a kind of knowledge is particularly area of research for the scientists, working in the field of Artificial Intelligence. There are various approaches to handle the imprecise knowledge, the most successful one is that of the Fuzzy logic, which was proposed by L.Zadeh, we discussed the same in our earlier sections of this unit.

In this section we will try to understand the Rough set theory approach, to manage the imprecise knowledge, it was proposed by Z. Pawlak. This theory is quite comprehensive and may be dealt as an independent discipline. It is quite connected with other theories and hence connected with various fields like AI, Machine Learning, Cognitive sciences, data mining, pattern recognition etc.

Rough set theory is quite comprehensive because of the following reasons :

• It requires no preliminary/additional information about the data as if it is the requirement of probability in statistics, or membership grades in the fuzzy set theory.

• Facilitates the user with efficient tools and techniques to detect the hidden patterns

• Promotes data reductionality i.e. it reduces the original data and, find minimal datasets from the data with the similar knowledge as it is in the original dataset.

• Helps to evaluate the data significance.

• Supports the mechanism to Sets the decision rules from the data, automatically

• It is easy to understand, best suited for concurrent or parallel or distributed processing , and offers straightforward interpretation of obtained results.

Following are the basic/elementary concepts of the Rough set theory :

1) Some information (data, knowledge) is associated with every object of the universe of discourse

2) Objects characterized by the same information are indiscernible or similar in view of the available information about them. The indiscernibility relation generated in this way is the mathematical basis of rough set theory. Any set of all indiscernible (similar) objects is called an elementary set, and forms a basic granule (atom) of knowledge about the universe.

3) Any union of some elementary sets is referred to as a crisp (precise) set – otherwise the set is rough (imprecise, vague).

4) Each rough set has boundary-line cases, i.e., objects which cannot be with certainty classified, by employing the available knowledge, as members of the set or its complement. Obviously rough sets, in contrast to precise sets, cannot be characterized in terms of information about their elements. With any rough set a pair of precise sets, called the lower and the upper approximation of the rough set, is associated.
Note: The lower approximation consists of all objects which surely belong to the set and the upper approximation contains all objects which possibly belong to the set. The difference between the upper and the lower approximation constitutes the boundary region of the rough set. Approximations are fundamental concepts of rough set theory.

5) Rough set based data analysis starts from a data table called a decision table, columns of which are labeled by attributes, rows – by objects of interest and entries of the table are attribute values.

6) Attributes of the decision table are divided into two disjoint groups called condition and decision attributes, respectively. Each row of a decision table induces a decision rule, which specifies decision (action, results, outcome, etc.) if some conditions are satisfied. If a decision rule uniquely determines decision in terms of conditions – the decision rule is certain. Otherwise the decision rule is uncertain.
Note: Decision rules are closely connected with approximations. Roughly speaking, certain decision rules describe lower approximation of decisions in terms of conditions, whereas uncertain decision rules refer to the boundary region of decisions.

7) With every decision rule two conditional probabilities, called the certainty and the coverage coefficient, are associated.

    a. The certainty coefficient expresses the conditional probability that an object belongs to the decision class specified by the decision rule, given it satisfies conditions of the rule.
    b. The coverage coefficient gives the conditional probability of reasons for a given decision. It turns out that the certainty and coverage coefficients satisfy Bayes' theorem. That gives a new look into the interpretation of Bayes' theorem, and offers a new method data to draw conclusions from data.

8.8 SUMMARY

In this unit the Fuzzy Systems are discussed along with the Introduction to Fuzzy Sets and their Representation. Later the conceptual understanding of Fuzzy Reasoning is build, and the same is used to perform the Fuzzy Inference. The unit finally discussed the concept of Rough Set Theory, also.

8.9 SOLUTIONS/ANSWERS

Check Your Progress - 1

Ex. 1: Discuss equality and subset relationship for the following fuzzy sets defined on the Universal set X = { a, b , c, d, e}

A = { a/.3, b/.6, c/.4 d/0, e/.7} ; B = {a/.4, b/.8, c/.9, d/.4, e/.7}; C = {a/.3, b/.7, c/.3, d/.2, e/.6}

SOLUTION: Both A and C are subsets of the fuzzy set B, because deg (x in A) ≤ deg (x in B) for all x ∈ X

Similarly degree (x in C) ≤ degree (x in B) for all x ∈ X

Further, A is not a subset of C, because, deg (c in A) = .4 > .3 = degree (c in C)

Also, C is not a subset of A, because, degree (b in C) = .7 > .6 = degree (b in A)

Check Your Progress - 2

Ex. 2: For the following fuzzy sets A = {a/.5, b/.6, c/.3, d/0, e/.9} and B = { a/.3, b/.7, c/.6, d/.3, e/.6}, find the fuzzy sets A ∩ B, A ∪ B and (A ∩ B)'

Solution : A ∩ B = {a/.3, b/.6, c/.3, d/0, e/.6},

where degree (x in A ∩ B) = min { degree (x in A), degree (x in B)}.

A ∪ B = {a/.5, b/.7, c/.6, d/.3, e/.9},

where degree (x in A ∪ B) = max {degree (x in A), degree (x in B)}.

The fuzzy set (A ∩ B)′ is obtained from A ∩ B, by the rule:

degree (x in (A ∩ B)′) = 1 − degree (x in A ∩ B).

Hence

(A ∩ B)′ = { a/.7, b/.4, c/.7, d/1, e/.4}

8.10 FURTHER READINGS

1. Ela Kumar, “ Artificial Intelligence”, IK International Publications
2. E. Rich and K. Knight, “Artificial intelligence”, Tata Mc Graw Hill Publications
3. N.J. Nilsson, “Principles of AI”, Narosa Publ. House Publications
4. John J. Craig, “Introduction to Robotics”, Addison Wesley publication
5. D.W. Patterson, “Introduction to AI and Expert Systems" Pearson publication