

# How Does Resolution Works in Propositional Calculus and Predicate Calculus, Introduction to Unification and Substitution

Dipanjan kumar Dey

Department of Computer Science & Engineering,  
Assistant professor of Prajnanananda Institute of  
Technology and Management (PITM)  
94/2, Park Street, Dist: Kolkata, Kolkata 700017,  
West Bengal (India)

## ABSTRACT

One of the most important rules of interference is resolution. Resolution basically works by using the principle of proof by contradiction. Propositional Resolution works only on expressions in clausal form. Before the rules of resolution can be applied, the premises and conclusions must be converted to this form. First part of this work consists of basic information about resolution like what are literals, clauses, empty clause, predicate, facts, rules, conjunctive normal form etc. What is it used for, what is their aim. One of the aims of this work is differentiating resolution between clauses-when clauses containing variables and When clauses containing no variables. When clauses containing no variables resolution are very easy and simple and then no need to substitution. When clauses containing variables resolution becomes complicated and then need a proper substitution through the cancellation of complementary literals. So substitution is an important role in resolution. Unification has been used in my work for performing resolution in predicate calculus. The unification algorithm tries to find out the Most General Unifier (MGU) between a set of atomic formulae. Theorem proving using resolution has also been included in my work which helps to solve many problems.

## Keywords

Resolution principle, Substitution, Unification, complementary literals Skolemization

## 1. INTRODUCTION

Resolution works on the principle of identifying complementary literals in two clauses and deleting them thereby forming a new literal (the resolvent). Propositional Resolution is a powerful rule of inference for Propositional Logic. Using Propositional Resolution (without axiom schemata or other rules of inference), it is possible to build a theorem prover that is sound and complete for all of Propositional Logic. One can perform Resolution from a Knowledge Base. A Knowledge Base is a collection of facts or one can even call it a database with all facts. Basically the logic process takes in some information called premises and produces some out puts called conclusions. Rules of resolution can be applied; the premises and conclusions must be converted to clausal form i.e. premises to predicate to clausal form, then resolution works. In this paper resolution works in propositional calculus when clauses containing no variables and resolution works in predicate calculus when clauses containing variables which are clearly explained with some examples.

## 2. STANDARD LOGIC SYMBOLS

Throughout this paper used the following standard logic symbols: “ $\rightarrow$ ” (implication), “ $\leftrightarrow$ ” (double implication or biconditional), “ $\sim$ ” (not), “ $\neg$ ” (not), “ $\wedge$ ” (and), “ $\vee$ ” (or), “ $\exists$ ” (there exists), “ $\forall$ ” (for all). If  $x$  is a variable then “ $\forall x$ ” is read as any one of- for all  $x$ , for each  $x$ , for every  $x$ . Again “ $\exists x$ ” is read as any one of- there exists  $x$ , for some  $x$ , for at least one  $x$ . The propositional calculus, from which we take all necessary properties of the logical operations  $\wedge$ ,  $\vee$ ,  $\neg$ ,  $\rightarrow$ , and  $\leftrightarrow$ , and the (first-order) predicate calculus, which to these propositional mechanisms adds compound functional and predicate constructions and the two quantifiers universal quantifier which is denoted by the symbol “ $\forall$ ” and existential quantifier which is denoted by the symbol “ $\exists$ ”.

## 3. LITERAL

A literal is either an atomic sentence or a negation of an atomic sentence.

For example, if  $p$  is a Logical constant, the following sentences are both literals  $P$  and  $\neg p$ . Another example:  $\sim p \vee \sim q \vee r$ , where  $p, q, r$  are literals.

## 4. CLAUSAL FORM

Propositional Resolution works only on expressions in clausal form. A clause is a disjunction of finitely many literals. For examples:  $\sim p \vee \sim q \vee r$  is a clause form, where  $p, q, r$  are literals.

## 5. EMPTY CLAUSE

A disjunction of no literals is called an empty clause and is denoted by  $\square$ . An empty clause shows that the negation of the conclusion is a complete contradiction, hence the negation of the conclusion is invalid or false or the assertion is completely valid or true.

## 6. UNIT CLAUSE

A Clause with one literal is known as unit clause. e.g.  $\{\sim q\}$ .

## 7. PROPOSITIONAL LOGIC OR PROPOSITIONAL CALCULUS

Propositional logic has only sentences, which represent facts. This is the simplest form of logic. Here all statements made are called propositions. A proposition in propositional logic takes only two values, either the proposition is true or it is false but not both or not partially true or false. The propositional calculus provides only the operations  $\wedge$ ,  $\vee$ ,  $\neg$ ,  $\rightarrow$ , and  $\leftrightarrow$  and the two constants ‘true’ and ‘false’.

## 8. PREDICATE LOGIC OR FIRST ORDER LOGIC OR PREDICATE CALCULUS

Predicate calculus is a generalization of propositional calculus. Some statements are there in real life situation which are correct but proposition logic fail to express them. To overcome this deficiency predicate logic has been introduced. For example: in real life statement- “All mammals suckle their young ones. Since elephant is a mammal, it suckles its young ones”. In this statement proposition logic fails to express it. The propositional logic is not powerful enough to represent all types of assertions that are used in computer science and mathematics, or to express certain types of relationship between propositions such as equivalence. For example, the assertion "x is greater than 1", where x is a variable, is not a proposition because you cannot tell whether it is true or false unless you know the value of x. Thus the propositional logic cannot deal with such sentences. However, such assertions appear quite often in mathematics and we want to do inference on those assertions.

## 9. PREDICATE

A predicate is defined as a relation that binds two atoms together for example: Ram likes aeroplanes. Here like is predicate that links two atoms “Ram” and “aeroplanes”. Symbolically likes (Ram, aeroplane). This predicate can be generalized as like (x, y) where x and y are variables meaning x likes y. Generally, predicates are used to describe certain properties or relationships between individuals or objects. A predicate can take arguments, which are terms. A predicate with one argument expresses a property of an object – Student (Bob) A predicate with no arguments is a simple proposition, as in propositional logic. Another example of an assignment is as follows. The domain consists of the four numbers 2, 3, 4, 5. The predicate “greater” is true if the first argument is greater than the second argument. Hence, greater (4, 3) is true and greater (3,4) is false. Assignment for the Predicate “greater”:

	2	3	4	5
2	0	0	0	0
3	1	0	0	0
4	1	1	0	0
5	1	1	1	0

Here, 0=false, 1= true.

In “John and Peter are brothers”, the phrase “are brothers” is a predicate. The entities connected this way, John and Peter are called terms. Terms play an important role in predicate calculus. A **term** is

1. a constant
2. a variable
3. If  $f$  is an  $n$ -place function, and  $t_1, \dots, t_n$  are terms, then  $f(t_1, \dots, t_n)$  is a term.

## 10. FACTS

Facts must start with a predicate which is an atom and ends with a full stop.

## 11. RULES

A rule is a predicate expression that uses Logical implication ( $\rightarrow$ ) to describe a relationship among facts.

For example (facts & rules)

$A = \{ \text{Fly}(x) \leftarrow \text{Bird}(x), \text{Bird}(\text{Tweety}) \leftarrow$

$\text{Bird}(x) \leftarrow \text{Penguin}(x), \text{Give\_egg}(x) \leftarrow \text{Bird}(x) \text{Penguin}(\text{Fred}) \leftarrow \text{Fly}(x) \leftarrow \text{Penguin}(x), \}$

Here basically two facts are there

1. Bird (Tweety)
2. Penguin(Fred)

And four types of rules are there:--

1.  $\text{Bird}(x) \leftarrow \text{Penguin}(x)$
2.  $\text{Gives\_egg}(x) \leftarrow \text{Bird}(x)$
3.  $\text{Fly}(x) \leftarrow \text{Penguin}(x)$
4.  $\text{Fly}(x) \leftarrow \text{Bird}(x)$

## 12. LIMITATIONS OF PROPOSITIONAL LOGIC

Many kinds of inference cannot be formalized in propositional logic. For example, most useful inferences involve applying a general rule to a specific case. But general-to-specific inferences like the following cannot be formalized in propositional logic

- All men are mortal
- Socrates is a man
- Therefore, Socrates is mortal.

This inference cannot be formalized in propositional logic because it refers to individual men, such as Socrates, and make generalizations about all men.

### 12.1 Normal Form In Propositional Logic

There are two major normal forms of propositional logic. One is conjunctive normal form (CNF) and the other is disjunctive normal form (DNF). Here discussing about CNF because in my work only CNF is required.

## 13. STEPS TO CONVERT TO CNF (CONJUNCTIVE NORMAL FORM)

Every sentence in Propositional Logic is logically equivalent to a conjunction of disjunctions of literals. A sentence expressed as a conjunction of disjunctions of literals is said to be in Conjunctive Normal Form (CNF).

### 1. Eliminate implications ‘ $\rightarrow$ ’

$A \rightarrow B = \sim A \vee B$

### 2. Eliminate biconditionals ‘ $\leftrightarrow$ ’

$A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A) = (\sim A \vee B) \wedge (\sim B \vee A)$

### 3. Reduce the not symbol by the formula $\sim(\sim A) = A$

### 4. DeMorgan’s Law

$\sim(A \wedge B) = \sim A \vee \sim B$  ..... DeMorgan’s Law

$\sim(A \vee B) = \sim A \wedge \sim B$  ..... DeMorgan’s Law

### 5. Eliminate AND ‘ $\wedge$ ’

$A \wedge B$  splits the entire clause into two separate clauses i.e. A and B

### 6. Use distributive law

$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$   $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

### For Example: convert $((p \rightarrow q) \rightarrow r)$ into CNF

The problem is  $((p \rightarrow q) \rightarrow r)$

$= \sim(p \rightarrow q) \vee r$ , since  $A \rightarrow B = \sim A \vee B$

$= \sim(\sim p \vee q) \vee r$

$= (p \wedge \sim q) \vee r$ , by DeMorgan’s Law

$= (p \vee r) \wedge (\sim q \vee r)$ ,

Use distributive law Thus  $(p \vee r) \wedge (\sim q \vee r)$

is the CNF of  $((p \rightarrow q)$

## 14. QUANTIFIERS

A quantifier is a symbol that permits one to declare or identify the range or scope of the variable in a logical expression.

There are two basic quantifiers used in logic one is universal quantifier which is denoted by the symbol “ $\forall$ ” and the other is existential quantifier which is denoted by the symbol “ $\exists$ ”. If  $x$  is a variable then “ $\forall x$ ” is read as any one of- for all  $x$ , for each  $x$ , for every  $x$ . Again “ $\exists x$ ” is read as any one of- there exists  $x$ , for some  $x$ , for at least one  $x$ .

For example:  $\forall x (A(x) \rightarrow B(x))$ , This formula states that for all  $x$ ,  $A(x)$  implies  $B(x)$ . In this formula universal quantifier “ $\forall$ ” applies over the entire formula  $(A(x) \rightarrow B(x))$ . Hence  $(A(x) \rightarrow B(x))$  is the scope of the quantifier.

**Note-1**  $\forall x P(x)$  means “for all  $x$ ,  $P$  of  $x$  is true”

Example:  $\forall x \text{ Happy}(x)$

If the universe of discourse is people, then this means that everyone is happy.

**Note-2:**  $\exists x P(x)$  means “there exists at least one  $x$  for which  $P$  of  $x$  is true”

Example:  $\exists x \text{ Happy}(x)$

If the universe of discourse is people, then this means there is at least one happy person.

Quantifiers indicate how frequently a certain statement is true. Specifically, the universal quantifier is used to indicate that a statement is always true, whereas the existential quantifier indicates that a statement is sometimes true. [14] Example-1: In “All cats have tails”, the word “all” indicates that the statement “cats have tails” is universally true. Example-2: If we say “Tweety is a bird”, we conclude that or common people will understand that Tweety flies. Now if we get further information about Tweety, it is not necessary that Tweety should fly because of a variety of reasons that Tweety is a penguin, Tweety’s wings are broken, Tweety is too weak to fly, Tweety is in caged, then we have to withdraw our previous conclusion and revise it by saying that Tweety doesn’t fly. This is existential quantifier, since existential quantifier indicates that a statement is sometimes true.

### 14.1 Relationship Between Universal And Existential Quantifiers

1.  $\forall x \neg P(x) \Leftrightarrow \neg \exists x P(x)$
2.  $\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$
3.  $\forall x P(x) \Leftrightarrow \neg \exists x \neg P(x)$
4.  $\exists x P(x) \Leftrightarrow \neg \forall x \neg P(x)$

**For examples:**

$\neg \exists y \text{ Happy}(y) \Leftrightarrow \forall y \neg \text{ Happy}(y)$

$\exists y \neg \text{ Happy}(y) \Leftrightarrow \neg \forall y \text{ Happy}(y)$

$\forall x \neg \text{ Likes}(x, \text{ John}) \Leftrightarrow \neg \exists x \text{ Likes}(x, \text{ John})$

$\forall x \text{ Likes}(x, \text{ John}) \Leftrightarrow \neg \exists x \neg \text{ Likes}(x, \text{ John})$

## 15. SOME PREDICATE CALCULUS AND THEIR ACTUAL MEANINGS

Calculus Expression	Actual Meaning
1. $\forall x(A(x))$	A is true for all $x$ .
2. $\exists x(A(x))$	A is true for some $x$ .
3. $\forall x(\sim A(x))$	A is false for all $x$ .
4. $\exists x(\sim A(x))$	A is false for some $x$ .
5. $\sim(\forall x(\sim A(x)))$	A is true for some $x$ .
6. $\sim(\exists x(\sim A(x)))$	A is true for all $x$ .

## 16. FREE AND BOUND VARIABLES

**Free variables:** A variable is free in a formula iff the occurrence is outside the scope of the quantifier having the variable. A variable is also free in a formula if at least one occurrence of it is free.

**Bound variables:** A variable is bound in a formula iff its occurrence is within the scope of the quantifier. A variable is also bound in situations where at least one occurrence of it is bound. [7]

Every occurrence of a variable  $x$  in a formula of the form  $\exists xB$  or of the form  $\forall xB$  is called bound occurrence, occurrence which are not bound are called free.

Note: Some formulas have no free variables they are called sentences.

For example:  $\forall x (A(x) \rightarrow B(x))$ , This formula states that for all  $x$ ,  $A(x)$  implies  $B(x)$ . In this formula universal quantifier “ $\forall$ ” applies over the entire formula  $(A(x) \rightarrow B(x))$ . Hence  $(A(x) \rightarrow B(x))$  is the scope of the quantifier. Any change in the quantifier has an effect on both  $A(x)$  and  $B(x)$ . So  $x$  is bound variable.

$\forall x \exists y (A(x, y, z)) \ \& \ \forall z(B(y, z))$ , in this formula  $z$  is free variable in  $\forall x \exists y (A(x, y, z))$  and  $z$  is bound variable in  $\forall z(B(y, z))$ .

## 17. NORMAL FORM IN PREDICATE LOGIC

Prenex normal form is the normal form in predicate logic. A formula “ $A$ ” in predicate logic is said to be prenex normal form if it has the form

$(Q_1x_1) (Q_2x_2) (Q_3x_3) \dots (Q_nx_n) B$ , Where  $(Q_ix_i)$  is either a  $\forall$  or  $\exists$  and  $B$  is formula without any quantifiers.  $B$  is called the matrix of the formula and  $(Q_1x_1) (Q_2x_2) (Q_3x_3) \dots (Q_nx_n)$  is called the prefix. i.e.  $Q_1, Q_2, Q_3, \dots, Q_n$  maybe either a universal quantifier “ $\forall$ ” or a existential quantifier “ $\exists$ ”.

## 18. CONVERTING TO PRENEX NORMAL FORM

**Step-1:** Involves bringing all quantifiers to the beginning of the formula  $(Q_i x_i) (M)$ ,  $i=1, 2, \dots, n$  Where,  $Q_i$  is either  $\forall$  (Universal Quantifier) or  $\exists$  (Existential Quantifier) and is called the prefix,  $M$  contains no Quantifiers and is called the matrix.

For example:  $\forall x \exists y (\sim A(x) \vee B(x, y))$  is a prenex normal form.  $\forall x \forall y \forall z ((A(x, y, z) \vee B(y, z)) \rightarrow C(x, z))$  is another prenex normal form.

**Example-1: Convert the formula  $\forall x (A(x) \rightarrow \exists y B(x, y))$  into prenex normal form. [7]**

Solution: The given formula is  $\forall x (A(x) \rightarrow \exists y B(x, y))$

$= \forall x (\sim A(x) \vee \exists y B(x, y))$ , since  $A \rightarrow B = \sim A \vee B$

$= \forall x \exists y (\sim A(x) \vee B(x, y))$  is the prenex normal form of  $\forall x (A(x) \rightarrow \exists y B(x, y))$ .

**Example-2 Convert the formula  $\sim((\forall x) p(x) \rightarrow (\exists y) (\forall z) q(y, z))$  into prenex normal form.**

Solution: The given formula is

$\sim((\forall x) p(x) \rightarrow (\exists y) (\forall z) q(y, z))$ .

$= \sim(\sim((\forall x) p(x)) \vee ((\exists y) (\forall z) q(y, z)))$ , since  $A \rightarrow B = \sim A \vee B$

$= ((\forall x) p(x)) \wedge \sim((\exists y) (\forall z) q(y, z))$ , since  $\sim(A \vee B) = \sim A \wedge \sim B$  and applying  $\sim(\sim A) = A$

$= ((\forall x) p(x)) \wedge ((\forall y) \sim((\forall z) q(y, z)))$ , since  $\sim(\exists x (A(x))) = \forall x (\sim A(x))$ .

$= (\forall x) p(x) \wedge (\forall y) (\exists z) \sim q(y, z)$ , since  $\sim (\forall x (A(x))) = \exists x (\sim A(x))$   
 $= (\forall x) (\forall y) (\exists z) p(x) \wedge \sim q(y, z)$ , is the required prenex normal form.

**Step-2:** In the required prenex normal form remove all ‘ $\leftrightarrow$ ’ and ‘ $\rightarrow$ ’ though  $\forall$  (Universal Quantifier) or  $\exists$  (Existential Quantifier) to the beginning of the formula.

For example:  $(\forall x) (\exists y) (\exists z) ((p(x, y) \vee \sim q(x, z)) \rightarrow r(x, t, z))$  is not the prenex normal form though  $\forall$  (Universal Quantifier) or  $\exists$  (Existential Quantifier) to the beginning of the formula.

**Convert the formula  $(\forall x) (\exists y) (\exists z) ((p(x, y) \vee \sim q(x, z)) \rightarrow r(x, y, z))$  into prenex normal form. [7]**

**Solution:** The given formula is  $(\forall x) (\exists y) (\exists z) ((p(x, y) \vee \sim q(x, z)) \rightarrow r(x, y, z))$ .

$= (\forall x) (\exists y) (\exists z) (\sim (p(x, y) \vee \sim q(x, z)) \vee r(x, y, z))$ , since  $A \rightarrow B = \sim A \vee B$ .

$= (\forall x) (\exists y) (\exists z) ((\sim p(x, y) \wedge q(x, z)) \vee r(x, y, z))$ , since  $\sim (A \vee B) = \sim A \wedge \sim B$  and applying  $\sim (\sim A) = A$ .

$= (\forall x) (\exists y) (\exists z) ((\sim p(x, y) \vee r(x, y, z)) \wedge (q(x, z) \vee r(x, y, z)))$ , Using distributive law,

This is the required prenex normal form.

## 19. SKOLEMIZATION

### Skolemization

Remove existentially quantified variables. Replace any existentially quantified variable  $\exists x$  that is in the scope of universally quantified variables  $\forall y_1 \dots \forall y_n$  with a new function  $F(y_1, \dots, y_n)$  (a **Skolem function**) Replace any existentially quantified variable  $\exists x$  that is not in the scope of any universally quantified variables with a new constant  $c$  (a **Skolem term**). [7]

Otherwise a formula  $P$  in the prenex normal form has a Skolem form  $P^S$  which is obtained when all  $\exists$  are removed by replacing the variables as functions of variables preceding it. If there is no  $\forall$  before  $\exists$  a constant symbol is used. Davis and Putnam method is widely used now-a-days.

The steps involved in this methodology are:

1. A formula of the first order logic can be transformed into prenex normal form where the matrix contains no quantifiers and the prefix is a sequence of quantifiers.
2. The matrix since it does not contain quantifiers can be transformed into a CNF.
3. Without affecting the inconsistency property, the existential quantifiers in the prefix can be eliminated by using Skolem functions.

**The effect of Skolemization:**

**Convert the formula  $\forall x \forall y \exists w \forall z Q(x, y, w, z, G(w, x))$  into its Skolem form.[7]**

**Solution:** The Skolem form for the given problem is  $\forall x \forall y \forall z Q(x, y, P(x, y), z, G(P(x, y), x))$  Where  $P$  is the Skolem function for  $w$ .

NB: the Skolem function is a function, so this is not decidable anymore.

**The effect of Skolemization:**

**Convert the formula  $P = \exists q \forall r \exists s \exists t (A(q, r) \rightarrow B(s, t))$  into its Skolem form**

**Solution:** In this formula  $\exists q$  is not preceded by any  $\forall$  quantifier.  $\exists s$  is preceded by an  $\forall$  quantifier and so is  $\exists t$ . Hence replace  $q$  by a constant say ‘a’,  $s$  and  $t$  by function of  $r$ . Therefore the Skolem form for the given problem is  $P^S = \forall r(A(a, r) \rightarrow B(f(r), g(r)))$ , where  $f$  and  $g$  are skolem functions for ‘ $r$ ’.

**The effect of Skolemization:**

**Convert the formula  $P = (\exists x) (\forall y) (\forall z) (\exists u) (\forall v) (\exists w) P(x, y, z, u, v, w)$  into its Skolem form.**

**Solution:** In this formula  $(\exists x)$  is preceded by no universal quantifiers, Therefore we replace the existential variable  $x$  by ‘a’.  $(\exists u)$  is preceded by  $(\forall y)$  and  $(\forall z)$ , Therefore we replace the existential variable  $u$  by a two- place function  $f(y, z)$ .  $(\exists w)$  is preceded by  $(\forall y)$ ,  $(\forall z)$  and  $(\forall v)$ . Therefore we replace the existential variable  $w$  by a three- place function  $g(y, z, v)$ . Therefore the Skolem form (standard form) for the given problem is  $P^S = (\forall y) (\forall z) (\forall v) P(a, y, z, f(y, z), v, g(y, z, v))$ , where  $f$  and  $g$  are skolem functions for  $(y, z)$  and  $(y, z, v)$  respectively.

**The effect of Skolemization:**

**Convert the formula  $P = (\forall x) (\exists y) (\exists z) ((\sim P(x, y) \wedge Q(x, z)) \vee R(x, y, z))$  into its Skolem form. [7]**

**Solution:** First the matrix is transformed into a CNF.  $(\forall x) (\exists y) (\exists z) ((\sim P(x, y) \vee R(x, y, z)) \wedge (Q(x, z) \vee R(x, y, z)))$  Now since  $(\exists y)$  and  $(\exists z)$  are both preceded by  $(\forall x)$ , the existential variables  $y$  and  $z$  are replaced respectively by one – place functions  $f(x)$  and  $g(x)$ , Therefore the Skolem form (standard form) for the given problem is  $P^S = (\forall x) ((\sim P(x, f(x)) \vee R(x, f(x), g(x))) \wedge (Q(x, g(x)) \vee R(x, f(x), g(x))))$ .

## 20. STEPS TO CONVERT PREDICATE LOGIC FORMULA INTO PRENEX NORMAL FORM

### 1. Eliminate implications ‘ $\rightarrow$ ’

$A \rightarrow B = \sim A \vee B$

### 2. Eliminate biconditionals ‘ $\leftrightarrow$ ’ $A \leftrightarrow B = (A \rightarrow B) \wedge (B \rightarrow A) = (\sim A \vee B) \wedge (\sim B \vee A)$

### 3. Reduce the not symbol by the formula $\sim (\sim A) = A$

### 4. DeMorgan’s Law

$\sim (A \wedge B) = \sim A \vee \sim B$  ..... DeMorgan’s Law

$\sim (A \vee B) = \sim A \wedge \sim B$  ..... DeMorgan’s Law

### 5. Eliminate AND ‘ $\wedge$ ’

$A \wedge B$  splits the entire clause into two separate clauses i.e.  $A$  and  $B$ .

### 6. Use distributive law

$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$   $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

### 7. Eliminate Existential Quantifier ‘ $\exists$ ’

To eliminate an independent Existential Quantifier, replace the variable by a Skolem constant. This process is called as Skolemization.

**For example:**  $\exists y$ : President ( $y$ )

Here ‘ $y$ ’ is an independent quantifier so we can replace ‘ $y$ ’ by any name (say –George Bush). So,  $\exists y$ : President ( $y$ ) becomes President (George Bush).

To eliminate a dependent Existential Quantifier we replace its variable by Skolem Function that accepts the value of ‘ $x$ ’ and returns the corresponding value of ‘ $y$ .’

**For example:**  $\forall x: \exists y$ : father of ( $x, y$ ) Here ‘ $y$ ’ is dependent on ‘ $x$ ’, so we replace ‘ $y$ ’ by  $S(x)$ . So,  $\forall x: \exists y$ : father\_of ( $x, y$ ) becomes  $\forall x: \exists y$ : father\_of ( $x, S(x)$ ). [13]

### 8. Eliminate Universal Quantifier ‘ $\forall$ ’

To eliminate the Universal Quantifier, drop the prefix in prenex normal form i.e. just drop  $\forall$  and the sentence then becomes in prenex normal form. [13]

### 9. Use the formula

$\sim (\forall x (A(x))) = \exists x (\sim A(x))$   $\sim (\exists x (A(x))) = \forall x (\sim A(x))$  Here negative signs bring before the atom.

## 21. STEPS FOR RESOLUTION

- Convert the given statements in Predicate/Propositional Logic
- Convert these statements into Conjunctive Normal Form
- Negate the Conclusion (Proof by Contradiction)
- Resolve using a Resolution Tree (Unification)

### 21.1 Resolution Principle

The resolution Principle is "Given any two clauses A and B, if there is a literal P1 in A which has a complementary literal P2 in B, delete P1 and P2 from A and B and construct a disjunction of the remaining clauses. The clause so constructed is called the resolvent of A and B." where A and B are parent clauses.[12]

### 21.2 Resolution on Propositional Calculus Resolution on propositional calculus

This type of problem clauses contains no variables. So resolution is very easy and simple. Resolution on propositional calculus works fine in situation where the result is either true or false but not both. Propositional Resolution works only on expressions in clausal form. Some statements are there in real life situation which are correct but proposition logic fail to express them. To overcome this deficiency predicate logic has been introduced.

For example in real life statement- "All mammals suckle their young ones. Since elephant is a mammal, it suckles its young ones". In this statement proposition logic fails to express it.

The idea of Propositional Resolution is simple. Suppose we have the clause  $\{p, q\}$ . In other words, we know that  $p$  is true or  $q$  is true. Suppose we also have the clause  $\{\neg q, r\}$ . In other words, we know that  $q$  is false or  $r$  is true. One clause contains  $q$ , and the other contains  $\neg q$ . If  $q$  is false, then by the first clause  $p$  must be true. If  $q$  is true, then, by the second clause,  $r$  must be true. Since  $q$  must be either true or false, then it must be the case that either  $p$  is true or  $r$  is true. So we should be able to derive the clause  $\{p, r\}$ .

The case we just discussed is an example. If we have the clause  $\{p, q\}$  and we also have the clause  $\{\neg q, r\}$ , then we can derive the clause  $\{p, r\}$  in a single step.

$\{p, q\}$   
 $\{\neg q, r\}$

---

$\{p, r\}$

### 21.3 Steps for Resolution On Propositional Calculus

**Step-1:** convert the entire proposition to clause form.

**Step-2:** negate  $p$  and convert the result to clause form.

**Step-3:** select two clauses say A&B call these the parent clauses.

**Step-4:** If there is a literal P1 in A which has a complimentary literal P2 in B delete P1&P2 from A&B and construct a disjunction of the remaining clauses. The clauses so constructed are called the resolvent of A & B.

**Step-5:** If the resolvent is empty clause then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

### 21.4 For Example (RESOLUTION FOR PROPOSITIONAL CALCULUS)

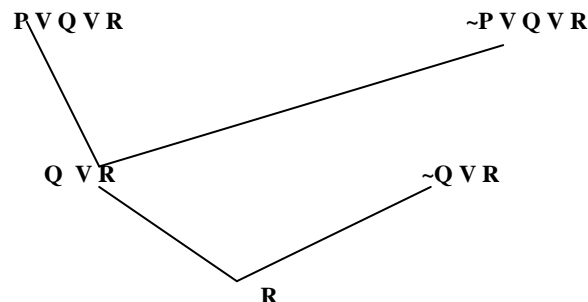
Consider the following clauses [10]

- A :  $P \vee Q \vee R$   
B :  $\neg P \vee Q \vee R$   
C :  $\neg Q \vee R$

Clauses A have the literal P which is complementary to  $\neg P$  in B. Hence both of them are deleted and a resolvent( disjunction of A and B after the complementary clauses are removed) is generated. That resolvent has again a literal Q whose negation is available in C. Hence resolving those two, one has the final resolvent.

- A :  $P \vee Q \vee R$  (given in the problem)  
B :  $\neg P \vee Q \vee R$  (given in the problem)  
D :  $Q \vee R$  (resolvent of A and B)  
C :  $\neg Q \vee R$  (given in the problem)  
E :  $R$  (resolvent of C and D)

It is possible to pectoris the path of the problem using a deduction tree.



### 21.5 Resolution In Propositional Logic When Clauses Containing No Variables, So Resolution Are Very Easy And Simple.

Let us consider the following clauses

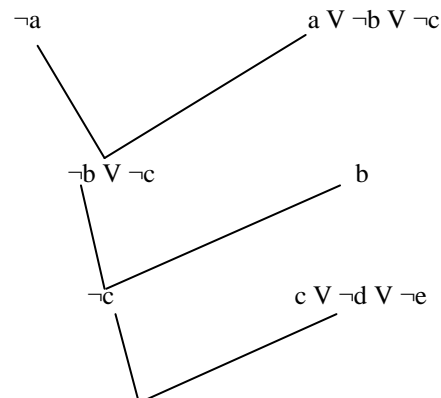
1.  $a \leftarrow b \wedge c$
2.  $b$
3.  $c \leftarrow d \wedge e$
4.  $e \vee f$
5.  $d \wedge \neg f$

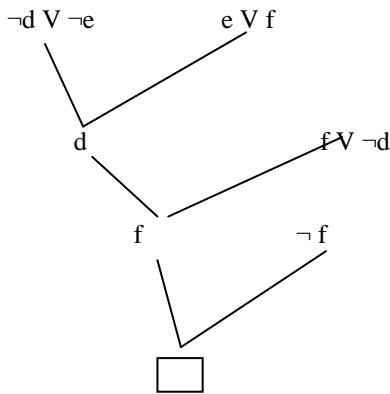
Convert the above predicates to clauses form


1.  $\sim (b \wedge c) \vee a = a \vee \sim (b \wedge c) = a \vee \sim b \vee \sim c$ , since  $A \rightarrow B = \sim A \vee B$ ,  $\sim (A \wedge B) = \sim A \vee \sim B$
2.  $b$
3.  $\sim (d \wedge e) \vee c = c \vee \sim (d \wedge e) = c \vee \sim d \vee \sim e$ , since  $A \rightarrow B = \sim A \vee B$ ,  $\sim (A \wedge B) = \sim A \vee \sim B$
4.  $e \vee f$
5.  $d$
6.  $\sim f$ , since  $A \wedge B$  splits the entire clause into two separate clauses i.e. A and B .

It is possible to pectoris the path of the problem using a deduction tree.

First, the goal to be proved, a, is negated and added to the clause set.





The derivation of  indicates that the database of clauses is inconsistent.

## 22. SUBSTITUTIONS

Substitutions are an essential part of interference process. When properly applied, they permit simplifications or the reduction of expressions through the cancellation of complementary literals. We say that two literals are complementary if they are identical but opposite in sign; i.e.  $P$  and  $\sim P$  are complementary literals. When clauses containing variables resolution becomes complicated and then need a proper substitution through the cancellation of complementary literals. So substitution is an important role in resolution. If  $A$  is a formula of predicate calculus, then  $(x/t)A$  denotes the formula that results when every occurrence of  $x$  in  $A$  is substituted by  $t$ . [12]. A substitution is defined as a set of pairs  $t_i / v_i$  where  $v_i$  are distinct variables and  $t_i$  are terms not containing the  $v_i$ . The  $t_i$  replace or are substituted for the corresponding  $v_i$  in any expression for which the substitution is applied. A set of substitutions  $\{ t_1 / v_1, t_2 / v_2, \dots, t_n / v_n \}$  where  $n \geq 1$  applied to an expression will be denoted by Greek letters  $\alpha, \beta$  and  $\delta$ . For example, If  $\beta = \{ a/x, g(b)/y \}$ , then applying  $\beta$  to the clause  $C = P(x, y) \vee Q(x, f(y))$  We obtain  $C' = C \beta = P(a, g(b)) \vee Q(a, f(g(b)))$ . [12]

There are three major types of substitutions, [10,7] viz,

1. Substitution of a variable by a constant.
2. Substitution of a variable by another variable.
3. Substitution of a variable by a function that does not contain the same variable.

For example:  $C_1 = \neg \text{dog}(x) \vee \text{animal}(x)$   $C_2 = \neg \text{animal}(y) \vee \text{die}(y)$  Resolvent:  $\neg \text{dog}(y) \vee \text{die}(y)$   $\{y/x\}$  Here  $x$  is substituted by  $y$ .

## 23. UNIFICATION

A substitution that makes two clauses resolvable is called a unifier and the process of identifying such unifiers is carried out by the unification algorithm. The unification algorithm tries to find out the Most General Unifier (MGU) between a set of atomic formulae. Given two expressions that are unifiable, such as expressions  $C_1$  and  $C_2$  with a unifier  $\beta$  with  $C_1 \beta = C_2$ , we say that  $\beta$  is a Most General Unifier (MGU) [12] if any other unifier  $\alpha$  is an instance of  $\beta$ . For example two unifier for literals  $P(u, b, v)$  and  $P(a, x, y)$  are  $\alpha = \{a/u, b/x, v/y\}$  and  $\beta = \{a/u, b/x, c/v, c/v\}$ . The former is a Most General Unifier (MGU) whereas the latter is not since it is an instance of the former.

To attempt to unify two literals, we first check if their initial predicate symbols are the same. If so, we can proceed,

otherwise there is no way that they can be unified, regardless of their arguments.

For example, two literals  $\neg P(x, y)$   $P(y, z)$  can be unified, since their initial predicate symbols  $P$  are the same.

Next we compare  $x$  and  $y$ , and decide that if we substitute  $y$  for  $x$ , they could match. We will write that substitution as  $y/x$ . What we need to do after finding the first substitution  $y/x$  is to make that substitution throughout the literals, giving  $P(y, y)$   $P(y, z)$  Now we can attempt to unify arguments  $y$  and  $z$ , which succeeds with the substitution  $z/y$ . The entire unification process has now succeeded with a substitution that is the composition of the two substitutions we found. We write the composition as  $(z/y)(y/x)$ . Note that the Most General Unifier (MGU) is  $[(z/y)(y/x)]$ .

For example, Ram likes aeroplanes. Here like is predicate and two atoms Ram and aeroplanes. Symbolically like (Ram, aeroplane). John hates aeroplanes. Here hate is predicate and two atoms John and aeroplanes. Symbolically hate (John, aeroplane). Thus two literals  $\neg \text{like}(\text{Ram}, \text{aeroplane})$   $\text{hate}(\text{John}, \text{aeroplane})$  cannot be unified, since their predicate symbols (like, hate) are not equal. So we can say two literals cannot be unified, so there is no chance for substitution.

Note-1: Unification can sometimes be applied to literals with the same single clause. When a Most General Unifier (MGU) exists such that two or more literals within a clause are unified, the clause remaining after deletion of all but one of the unified literals is called a factor of the original clause. Thus, given a clause  $C = P(x) \vee Q(x, y) \vee P(f(z))$  the factor  $C' = C \beta = P(f(z)) \vee Q(f(z), y)$  is obtained where  $\beta = \{ f(z)/x \}$ .

Note-2: The basic idea of Unification is very simple. Any substitution that makes two or more expressions equal is called a unifier for the expressions.

Note-3: The substitution  $(a_1/a_2, a_3/a_4, \dots)(b_1/b_2, b_3/b_4, \dots)$  means to apply all the substitutions of the right-most list, then take the result and apply all the ones of the next list, and so forth, until all substitutions have been applied.

Note-4: The literals

hate (x, y)  
hate (John, z)

could be unified with any of the following substitutions:

(John/x, z/y)  
(John/x, y/z)  
(John/x, Peter/y, Peter/z).

**Example:** Find the Most General Unifier (MGU) of

$C(x, f(g(x)), a)$  and  $C(b, y, z)$

**Solution:**

$C(x, f(g(x)), a)$  and  $C(b, y, z)$  SUBST = [ ]

$C(x, f(g(x)), a)$  and  $C(b, y, z)$  SUBST = [(x/b)]

$C(b, y, a)$  and  $C(b, y, z)$  SUBST = [(y/ f(g(x)), (x/b)]

$C(b, y, z)$  and  $C(b, y, z)$  SUBST = [(z/a), (y/ f(g(x)), (x/b)]

The SUBST started with an empty substitution and has made some substitutions that unifies both the clauses. Hence the Most General Unifier (MGU) is  $[(z/a), (y/ f(g(x)), (x/b)]$ .

## 24. RESOLUTION ON THE PREDICATE CALCULUS

Substitution and unification are used for performing resolution in predicate calculus.

A literal and its negation in parent clauses produce a resolvent only if they unify under some substitution  $\sigma$ .  $\Sigma$  is then applied to the resolvent before adding it to the clause set.

C1=  $\neg \text{dog}(x) \vee \text{animal}(x)$

C2=  $\neg \text{animal}(y) \vee \text{die}(y)$

Resolvent :  $\neg \text{dog}(y) \vee \text{die}(y) \{y/x\}$

Here x is substituted by y.

## 25. STEPS FOR RESOLUTION ON PREDICATE CALCULUS

**Step-1:** convert all the statements to clause form.

**Step-2:** negate p and convert the result to clause form.

**Step-3:** select two clauses say A&B call these the parent clauses

**Step-4:** Resolve them together The resolvent will be the disjunction of all the literals of both the parent clauses with appropriate substitutions performed and with the following exceptions .If there is one pair of literals L1 and  $\neg L2$  such that one of the parent clauses contain L2 and the other contains L1 and if L1 and L2 are unifiable then neither L1 nor L2 should appear in the resolvent.If there is more than one pair of complimentary literals, only one pair should be omitted from the resolvent.

**Step-5:** If the resolvent is empty clause then a contradiction has been found. If it is not, then add it to the set of clauses available to the procedure.

## 26. THEOREM PROVING USING RESOLUTION

There are mostly two basic methods of theorem proving. [10]  
**Method-1:** start with the given axioms, use the rules of inference and then prove the theorem.

**Method-2:** prove the negation of the result cannot be true.

Note: The second method is commonly known as theorem proving using resolution.

The following steps are the solving problem using theorem proving using resolution:

**Step-1:** Find the negation of the result to be proved.

**Step-2:** Add it as a valid statement to the given set of statements.

**Step-3:** Perform resolution on these statements until a contradiction is encountered.

**Step-4:** Conclude that the contradiction is due to the assumed negation of the result.

**Step-5:** So the negated assumption that is made is false or the result to be proved is true.

## 27. SOLVING REAL LIFE EXAMPLE OF THEOREM PROVING USING RESOLUTION:

**Illustration- 1: Let us consider the following statements—**  
“Tom is a dog” “All dogs are animal” And “animals will die” Prove that “Tom will die”.

**Solution:** Here the premises are “Tom is a dog” “All dogs are animal” And “animals will die”

Convert the premises to predicate –

The predicate forms are

1.  $\forall(x) (\text{dog}(x) \rightarrow \text{animal}(x))$

2.  $\text{dog}(\text{Tom})$

3.  $\forall(y) (\text{animal}(y) \rightarrow \text{die}(y))$

In this problem basically one fact which is  $\text{dog}(\text{Tom})$  and two rules are there in this problem they are

1.  $\text{dog}(x) \rightarrow \text{animal}(x)$

2.  $\text{animal}(y) \rightarrow \text{die}(y)$

Now convert the predicates to clause form by applying  $a \rightarrow b = \neg a \vee b$

1.  $\neg \text{dog}(x) \vee \text{animal}(x)$

2.  $\text{dog}(\text{Tom})$

3.  $\neg \text{animal}(y) \vee \text{die}(y)$

By applying theorem proving using resolution, of method-2 as in above,

First let us assume negation of the result

$\neg \text{die}(\text{Tom})$  (1)

The given axioms are-

$\text{dog}(\text{Tom})$  (2)  $\forall(x)$

$(\text{dog}(x) \rightarrow \text{animal}(x))$  (3)  $\forall(y)$

$(\text{animal}(y) \rightarrow \text{die}(y))$  (4)

Equations (3) & (4) can be written as

$\neg \text{dog}(x) \vee \text{animal}(x)$  (5)

$\neg \text{animal}(y) \vee \text{die}(y)$  (6)

In equations (5) & (6), substitute  $x = \text{Tom}$ ,  $y = \text{Tom}$ , gives,

$\neg \text{dog}(\text{Tom}) \vee \text{animal}(\text{Tom})$

(7)

$\neg \text{animal}(\text{Tom}) \vee \text{die}(\text{Tom})$  (8)

Resolving equations (7) & (8), gives,

$\neg \text{dog}(\text{Tom}) \vee \text{die}(\text{Tom})$

(9)

Resolving

equations (1) & (9), gives,

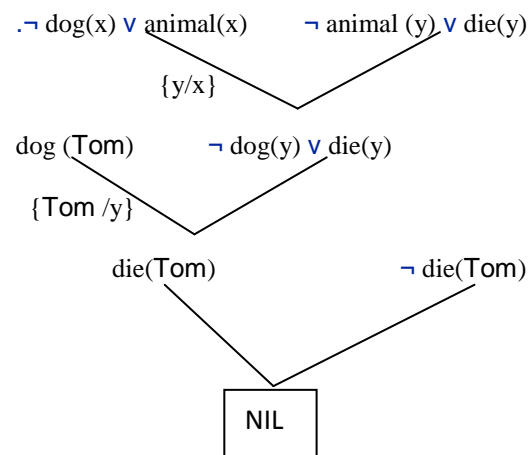
$\neg \text{dog}(\text{Tom})$

(10) Resolving equations (2) & (10), have a

contradiction.

This contradiction was due to the assumption that was made, i.e. the negation of the result. Hence the negation of the result is false or the result is true.Hence proved that is Tom will die.

By tree diagram--



Hence we see that the negation of the conclusion has been proved as a complete contradiction with the given set of facts. Hence the negation is completely invalid or false or the assertion is completely valid or true. Hence Tom will die (Proved).

**Illustration- 2: Let the given theory be**

$A = \{\text{Fly}(x) \leftarrow \text{Bird}(x),$

$\text{Bird}(\text{Tweety}) \leftarrow$

$\text{Bird}(x) \leftarrow \text{Penguin}(x),$

$\text{Give\_egg}(x) \leftarrow \text{Bird}(x)$

$\text{Penguin}(\text{Fred}) \leftarrow$

$\text{Fly}(x) \leftarrow \text{Penguin}(x), \}$

Can Tweety fly? Write the success set.

**Solution: ----** Negate the conclusion gives,  $\sim \text{Fly}(\text{Tweety})$

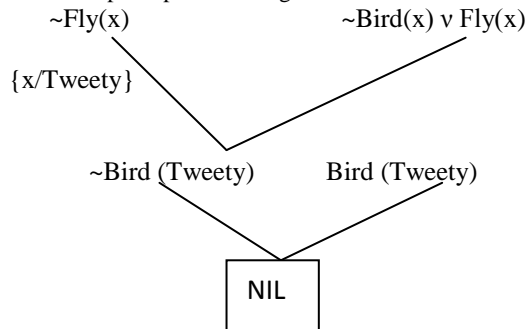
From the given problem, gives,

$\sim \text{Bird}(x) \vee \text{Fly}(x)$



~ Penguin(x) v Bird(x)  
~ Bird(x) v Gives\_egg(x)  
~ Penguin(x) v Fly(x)  
Bird ( Tweety )  
Penguin (Fred)  
The ground atoms are  
{ Bird ( Tweety )  
Fly (Tweety)  
Penguin (Tweety)  
Gives\_egg(Tweety)  
~ Fly (Tweety)  
Penguin (Fred)  
Bird (Fred)  
Gives\_egg(Fred)  
~Fly (Fred)  
Fly (Fred)}

Using resolution principle and using a deduction tree we have,



Hence we see that the negation of the conclusion has been proved as a complete contradiction with the given set of facts. Hence the negation is completely invalid or false or the assertion is completely valid or true. Therefore Tweety can fly.

Success set: - { Bird(x), Fly(x), ~ Fly(x) }

Note: Success Set is a subset of ground atom.

Constant function - Tweety, Fred.

### Illustration – 3:

Now let us see another example which uses theorem proving using resolution.

**Problem Statement are:**

1. John likes all kind of food.
2. Apples and chicken are food.
3. Anything anyone eats and is not killed by is food.
4. Peter eats peanuts and is still alive.
5. Robin eats everything that Peter eats.

Translate these sentences into formulas in predicate logic.

Convert the formulas into clause form.

Prove that John likes peanuts using resolution. [11]

**Solution:**

**Step 1:** Converting the given statements into Predicate/Propositional Logic

1.  $\forall x: \text{food}(x) \rightarrow \text{likes}(\text{John}, x)$
2.  $\text{food}(\text{apple}) \wedge \text{food}(\text{chicken})$
3.  $\forall a: \forall b: \text{eats}(a, b) \wedge \text{killed}(a) \rightarrow \text{food}(b)$
4.  $\text{eats}(\text{Peter}, \text{Peanuts}) \wedge \text{alive}(\text{Peter})$
5.  $\forall c: \text{eats}(\text{Peter}, c) \rightarrow \text{eats}(\text{Robin}, c)$
6.  $\forall d: \text{alive}(d) \rightarrow \sim \text{killed}(d)$
7.  $\forall e: \sim \text{killed}(e) \rightarrow \text{alive}(e)$

**Conclusion:** likes (John, Peanuts)

**Step 2:** Convert into CNF

- i.  $\sim \text{food}(x) \vee \text{likes}(\text{John}, x)$
- ii. Food (apple)
- iii. Food (chicken)

iv.  $\sim \text{eats}(a, b) \vee \text{killed}(a) \vee \text{food}(b)$

v. Eats (Peter, Peanuts)

vi. Alive (Peter)

vii.  $\sim \text{eats}(\text{Peter}, c) \vee \text{eats}(\text{Robin}, c)$

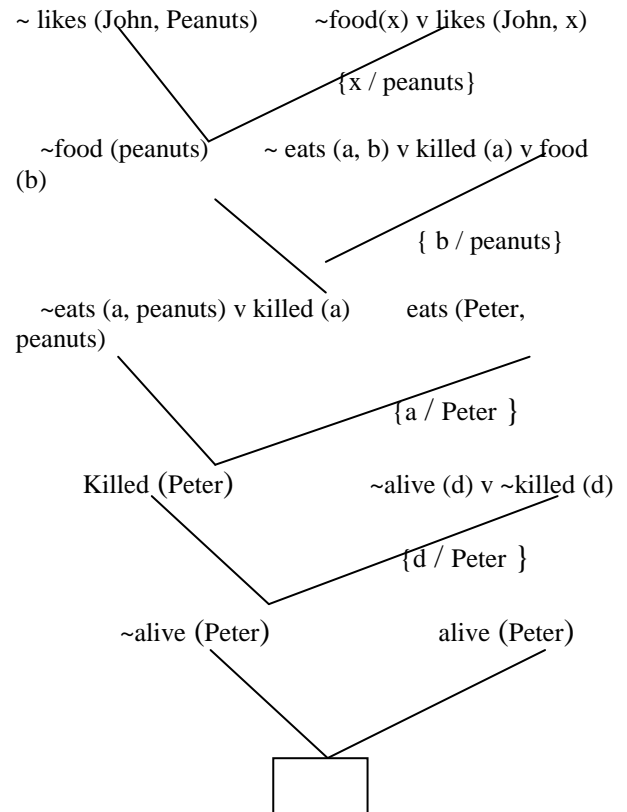
viii.  $\sim \text{alive}(d) \vee \sim \text{killed}(d)$

ix. Killed (e) v alive (e)

**Conclusion:** likes (John, Peanuts)

**Step 3:** Negate the conclusion ~ likes (John, Peanuts)

**Step 4:** Resolve using a resolution tree



Hence we see that the negation of the conclusion has been proved as a complete contradiction with the given set of facts. Hence the negation is completely invalid or false or the assertion is completely valid or true. Hence Proved.

Now the following simple example will show clearly how two methods (explained above) help in theorem proving.

Let us consider the following statements—

Given that

1)  $\forall(x) [\text{physician}(x) \rightarrow \text{knows\_surgery}(x)]$

2) Physician (John)

Prove that knows\_surgery(John)

**Proof:**

**By using Method-1 of theorem proving:**

Using Modus ponens- Modus ponens states that if there is an axiom of the form  $P \rightarrow Q$  and another of the form  $P$ , then  $Q$  logically follows. Here in the given problem, assuming Physician (John) as  $P$  and  $[\text{physician}(x) \rightarrow \text{knows\_surgery}(x)]$  as  $Q$ . Substitute  $x = \text{John}$ , the result  $\text{knows\_surgery}(\text{John})$  logically follows.

**By using Method-2 of theorem proving:**

Let us assume the negation of the result

$\neg \text{knows\_surgery}(\text{John})$  (1)

The given axioms are

Physician (John) (2)

$\forall(x) [\text{physician}(x) \rightarrow \text{knows\_surgery}(x)]$  (3)

Equation (3) can be written as



$$\neg \text{Physician}(x) \vee \text{knows\_surgery}(x) \quad (4)$$

[The quantifier is universal. If it had been existential, then skolem function has to be used]

Substitute  $x = \text{John}$ , equation (4) becomes,  

$$\neg \text{Physician}(\text{John}) \vee \text{knows\_surgery}(\text{John}) \quad (5)$$

Resolving equations (1) & (5), gives,  

$$\neg \text{Physician}(\text{John}) \quad (6)$$

Resolving equations (2) & (6), have a contradiction. This contradiction was due to the assumption that was made, i.e. the negation of the result. Hence the negation of the result is false or the result is true.

Thus  $\text{knows\_surgery}(\text{John})$  (proved).

## 28. DEFICIENCIES OF PREDICATE LOGIC AND RESOLUTION:

Since logic base system and theorem proving techniques are monotonic in nature, i.e. if a proposition is made which is true, it remain true under all circumstances. This is monotonic reasoning system. Monotonic reasoning system cannot work in real life environment. All the theorems are proved by this methodology only, but in real life or real world are never monotonic since (1) situation changes time to time (2) information available is always incomplete. This is called nonmonotonic reasoning system. So logic base system and theorem proving techniques cannot work efficiently in nonmonotonic reasoning system. Some difficulty with the use of theorem proving in AI systems is that there are some kinds of information that are not easily represented in predicate logic, like some example [11] "It is very hot today." How can relative degrees of heat be represented? "Blond haired people often have blue eyes." How can the amount of certainty be represented? Since logic base system does not provide facilities for handling uncertainty. Every information logic deals have to either true or false but never both or partially true or false. For example-If we say Tweety is a bird, we conclude that or common people will understand that Tweety flies. Now if we get further information about Tweety, it is not necessary that Tweety should fly because of a variety of reasons that Tweety is a penguin, Tweety's wings are broken, Tweety is too weak to fly, Tweety is in caged, then we have to withdraw our previous conclusion and revise it by saying that Tweety doesn't fly. In this situation predicate logic and resolution doesn't work.

## 29. RESOLUTION EXAMPLE (REAL LIFE)

Anyone passing his science exams and winning the lottery is happy. But anyone who studies or is lucky can pass all his exams. John did not study but John is lucky. Anyone who is lucky wins the lottery. Is John happy? Translate these sentences into formulas in predicate logic. Convert the formulas into clause form.

**Solution:** Here the premises are-

1. Anyone passing his history exams and winning the lottery is happy.
2. But anyone who studies or is lucky can pass all his exams.
3. John did not study, but John is lucky
4. Anyone who is lucky wins the lottery.

**Convert to predicate logic**

1. Anyone passing his history exams and winning the lottery is happy.  
 $\forall x \text{Pass}(x, \text{science}) \wedge \text{Win}(x, \text{Lottery}) \rightarrow \text{Happy}(x)$
2. But anyone who studies or is lucky can pass all his exams.  
 $\forall x \forall y \text{Study}(x) \vee \text{Lucky}(x) \rightarrow \text{Pass}(x,y)$
3. John did not study, but John is lucky

$$\neg \text{Study}(\text{John}) \wedge \text{Lucky}(\text{John})$$

4. Anyone who is lucky wins the lottery.

$$\forall x \text{Lucky}(x) \rightarrow \text{Win}(x, \text{Lottery})$$

**Convert to CNF**

Eliminate implications:

$$1. \forall x \neg (\text{Pass}(x, \text{science}) \wedge \text{Win}(x, \text{Lottery})) \vee \text{Happy}(x)$$

$$2. \forall x \forall y \neg (\text{Study}(x) \vee \text{Lucky}(x)) \vee \text{Pass}(x,y)$$

$$3. \neg \text{Study}(\text{John}) \wedge \text{Lucky}(\text{John})$$

$$4. \forall x \neg \text{Lucky}(x) \vee \text{Win}(x, \text{Lottery})$$

Move  $\neg$  inward

$$1. \forall x \neg \text{Pass}(x, \text{science}) \vee \neg \text{Win}(x, \text{Lottery}) \vee \text{Happy}(x)$$

$$2. \forall x \forall y (\neg \text{Study}(x) \wedge \neg \text{Lucky}(x)) \vee \text{Pass}(x,y)$$

$$3. \neg \text{Study}(\text{John}) \wedge \text{Lucky}(\text{John})$$

$$4. \forall x \neg \text{Lucky}(x) \vee \text{Win}(x, \text{Lottery})$$

Standardize variables: no action needed.

Move quantifiers left: no action needed except drop quantifiers Skolemize: no action needed.

Distribute  $\wedge$  over  $\vee$

$$1. \neg \text{Pass}(x, \text{science}) \vee \neg \text{Win}(x, \text{Lottery}) \vee \text{Happy}(x)$$

$$2. (\neg \text{Study}(x) \vee \text{Pass}(x,y)) \wedge (\neg \text{Lucky}(x) \vee \text{Pass}(x,y))$$

$$3. \neg \text{Study}(\text{John}) \wedge \text{Lucky}(\text{John})$$

$$4. \neg \text{Lucky}(x) \vee \text{Win}(x, \text{Lottery})$$

Flatten nested conjunctions and disjunctions

no action necessary.

State as a set of disjunction of literals

$$1. \neg \text{Pass}(x, \text{science}) \vee \neg \text{Win}(x, \text{Lottery}) \vee \text{Happy}(x)$$

$$2. a. \neg \text{Study}(x) \vee \text{Pass}(x,y)$$

$$2. b. \neg \text{Lucky}(x) \vee \text{Pass}(x,y)$$

$$3. a. \neg \text{Study}(\text{John})$$

$$3. b. \text{Lucky}(\text{John})$$

$$4. \neg \text{Lucky}(x) \vee \text{Win}(x, \text{Lottery})$$

Standardize variables apart

$$1. \neg \text{Pass}(x1, \text{science}) \vee \neg \text{Win}(x1, \text{Lottery}) \vee \text{Happy}(x1)$$

$$2. a. \neg \text{Study}(x2) \vee \text{Pass}(x2,y1)$$

$$2. b. \neg \text{Lucky}(x3) \vee \text{Pass}(x3,y2)$$

$$3. a. \neg \text{Study}(\text{John})$$

$$3. b. \text{Lucky}(\text{John})$$

$$4. \neg \text{Lucky}(x4) \vee \text{Win}(x4, \text{ottery})$$

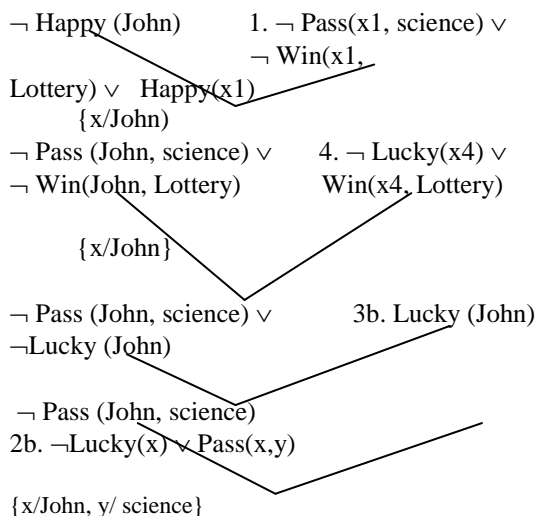
Now In Conjunctive Normal Form (CNF) Resolution Proof

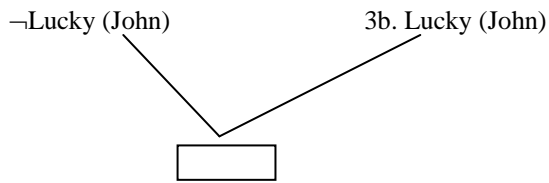
Procedure • Assert negation of goal – In this case the goal is to prove

Happy(John) – Add the clause  $\neg \text{Happy}(\text{John})$  to the KB •

Resolve clauses together until FALSE is derived.

**Resolution Proof Tree**





Hence the negation of the conclusion has been proved as a complete contradiction with the given set of facts. Hence the negation is completely invalid or false or the assertion is completely valid or true. Hence Proved, i.e. Happy (John)

### 30. CONCLUSION AND FUTURE WORK

Resolution using refutation is much simpler than the method using the rules of inference. Proving theorems is considered to require high intelligence. If knowledge is represented by logic, theorem proving is reasoning. Once the theorem prover shows that the negated goal is inconsistent with the given set of axioms, it follows that the original goal must be consistent. Propositional Resolution can be used in a proof procedure that always terminates without losing completeness. I have shown in my work how does unification and substitution are related to each other. I have also shown in my work clearly when two literals are unifiable. Two literals unify if their predicate symbols are same, otherwise there is no way they can be unified. When two literals are not unified then substitution cannot apply. The main significant of substitution that makes two literals are identical. When two literals are identical and complementary then they resolve and resolution principle works finely. Predicate calculus is increasingly used for specifying the requirements of computer applications. In the area of proving program correctness, predicate calculus allows one to precisely state under which conditions a program gives the correct output.

### 31. ACKNOWLEDGEMENTS

Author of this paper sincerely thank to Prof. Kumar Sankar Ray, Dept of Electronics & Communication Science Unit, Indian Statistical Institute, Kolkata- 700108, India, for his constant support and encouragement.

### 32. REFERENCES

- [1] [Pastre2002] D. Pastre, Strong and weak points of the MUSCADET theorem prover, AI Communications, 15(2-3):147-160, 2002, [http://www.cs.miami.edu/\\_tptp](http://www.cs.miami.edu/_tptp).
- [2] [Pastre1993] D. Pastre, Automated Theorem Proving in Mathematics, Annals on Artificial Intelligence and Mathematics, 8(3-4):425-447, 1993.

- [3] [Robinson1965] J.A. Robinson, A machine oriented logic based on the resolution principle, J.ACM12:23-41, 1965.
- [4] [Bledsoe1977] W. W. Bledsoe, Non-Resolution Theorem Proving, Journal of Artificial Intelligence, 9:1-35, 1977.
- [5] Artificial Intelligence: A Modern Approach by Stuart Russell and Peter Norvig, Second Edition, Published 2003 Prentice Hall .
- [6] [Jec97] Jech, T.J.: Set Theory, 2<sup>nd</sup> edn. Perspectives in Mathematical Logic. Springer, Berlin (1997).
- [7] Chang C and Lee R, Symbolic Logic Mechanical Theorem proving, Academic press, New York, 1973.
- [8] Nilsson N Principal of Artificial Intelligence, Tioga Publishing Company, 1980.
- [9] Winston P H Artificial Intelligence, 2nd edition, Addison-Wesley, Menlo-Park, California, 1984.
- [10] Foundation of artificial intelligence and expert system by Janakiraman.
- [11] Artificial intelligence, 3<sup>rd</sup> edn, by Elaine Rich ,Kevin Knight, Shivashankar B Nair.
- [12] Introduction to 'Artificial intelligence and expert system' by Dan W. Patterson, PHI
- [13] Resolution By Ankit Shah, Professor Harper Langston Discrete Mathematics Summer 2007
- [14] Artificial intelligence notes- reasoning methods, lecturer: Coşkun Sönmez

### 33. AUTHOR'S PROFILE

Dipanjan Kumar Dey, graduated from Calcutta University, India. M.sc (Mathematics) and M.Tech (Computer Science & Engineering) from M.C.K.V Institute of Engineering (under West Bengal University & Technology, India). He is currently Assistant Professor of Mathematics & Computer Science in Prajnanananda Institute of Technology & Management, West Bengal, India. He is also Faculty member of Institute of Chartered financial Analysis of India (ICFAI) and Academic Counselor, Assistant Coordinator of Indira Gandhi National Open University (IGNOU) study center 2804, Kolkata, India. He is a Science Journalist having Post Graduate certificate course on journalism and media practice from National Council for the Science and Technology Communication, GOVT. OF INDIA, New Delhi. Mr. Dey has to his credit a significant number of research papers published in international journals of repute. His research interests in Genetic Algorithms, Soft Computing, Fuzzy Set, Artificial intelligence, Mobile computing.