Resolution Method in AI

Resolution Method in AI

Resolution method is an inference rule which is used in both Propositional as well as First-order Predicate Logic in different ways. This method is basically used for proving the satisfiability of a sentence. In resolution method, we use **Proof by Refutation** technique to prove the given statement.

The key idea for the resolution method is to use the knowledge base and negated goal to obtain null clause(which indicates contradiction). Resolution method is also called **Proof by Refutation**. Since the knowledge base itself is consistent, the contradiction must be introduced by a negated goal. As a result, we have to conclude that the original goal is true.

Resolution Method in Propositional Logic

In propositional logic, resolution method is the only inference rule which gives a new clause when two or more clauses are coupled together.

Using propositional resolution, it becomes easy to make a theorem prover sound and complete for all.

The process followed to convert the propositional logic into resolution method contains the below steps:

- Convert the given axiom into clausal form, i.e., disjunction form.
- Apply and proof the given goal using negation rule.
- Use those literals which are needed to prove.

But, before solving problems using Resolution method, let's understand two normal forms

Conjunctive Normal Form(CNF)

In propositional logic, the resolution method is applied only to those clauses which are disjunction of literals. **There are following steps used to convert into CNF:**

1) Eliminate bi-conditional implication by replacing A ⇔ B with (A → B) Λ (B →A)

2) Eliminate implication by replacing A → B with ¬A V B.

3) In CNF, negation(¬) appears only in literals, therefore we move it inwards as:

- **¬ (¬A) ≡ A** (double-negation elimination)
- **¬ (A Λ B) ≡ (¬A V ¬B)** (De Morgan)
- **¬(A V B) ≡ (¬A Λ ¬B)** (De Morgan)

4) Finally, using distributive law on the sentences, and form the CNF as:

$(A_1 \lor B_1) \land (A_2 \lor B_2) \land \ldots. \land (A_n \lor B_n)$.

Note: CNF can also be described as AND of ORS

Disjunctive Normal Form (DNF)

This is a reverse approach of CNF. The process is similar to CNF with the following difference:

$(A_1 \land B_1) \lor (A_2 \land B_2) \lor \ldots \lor (A_n \land B_n)$. In DNF, it is OR of ANDS, a sum of products, or a cluster concept, whereas, in CNF, it is ANDs of Ors.

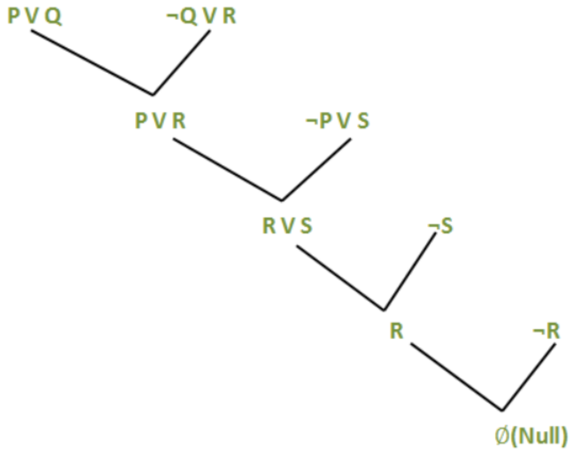
Applying resolution method:

In (2), Q → R will be converted as (¬Q V R)

In (3), P → S will be converted as (¬P V S)

Negation of Goal (¬R): It will not rain.

Finally, apply the rule as shown below:



After applying Proof by Refutation (Contradiction) on the goal, the problem is solved, and it has terminated with a **Null clause (Ø).** Hence, the goal is achieved. Thus, It is not raining.

Note: We can have many examples of Proposition logic which can be proved with the help of Propositional resolution method.

Resolution Method in FOPl/Predicate Logic

Resolution method in FOPL is an uplifted version of propositional resolution method.

In FOPL, the process to apply the resolution method is as follows:

- Convert the given axiom into CNF, i.e., a conjunction of clauses. Each clause should be dis-junction of literals.
- Apply negation on the goal given.
- Use literals which are required and prove it.
- Unlike propositional logic, FOPL literals are complementary if one unifies with the negation of other literal.

For example: {Bird(F(x)) V Loves(G(x), x)} and {¬Loves(a, b) V ¬Kills(a, b)}

Eliminate the complementary literals Loves(G(x),x) and Loves(a,b)) with θ ={a/G(x), v/x} to give the following output clause:

{Bird(F(x)) V ¬Kills(G(x),x)}

The rule applied on the following example is called **Binary Resolution** as it has solved exactly two literals. But, binary resolution is not complete. An alternative approach is to extend the **factoring** i.e., to remove redundant literals to the first order case. Thus, the combination of binary resolution and factoring is complete.

Conjunctive Normal Form

There are following steps used to convert into CNF:

- Eliminate the implications as:

∀x: A(x) → B(x) with {¬ x: ¬A(∀x) V B(x)}

- Move negation (¬) inwards as:

¬∀x: A becomes ∃x: ¬A and,

Example OF Propositional Resolution

Consider the following Knowledge Base:

1. The humidity is high or the sky is cloudy.
2. If the sky is cloudy, then it will rain.
3. If the humidity is high, then it is hot.
4. It is not hot.

Goal: It will rain.

Use propositional logic and apply resolution method to prove that the goal is derivable from the given knowledge base.

Solution: Let's construct propositions of the given sentences one by one:

1. Let, P: Humidity is high.

Q: Sky is cloudy.

It will be represented as **P V Q.** 1

2) Q: Sky is cloudy. **...from(1)**

R

Let, R: It will rain.

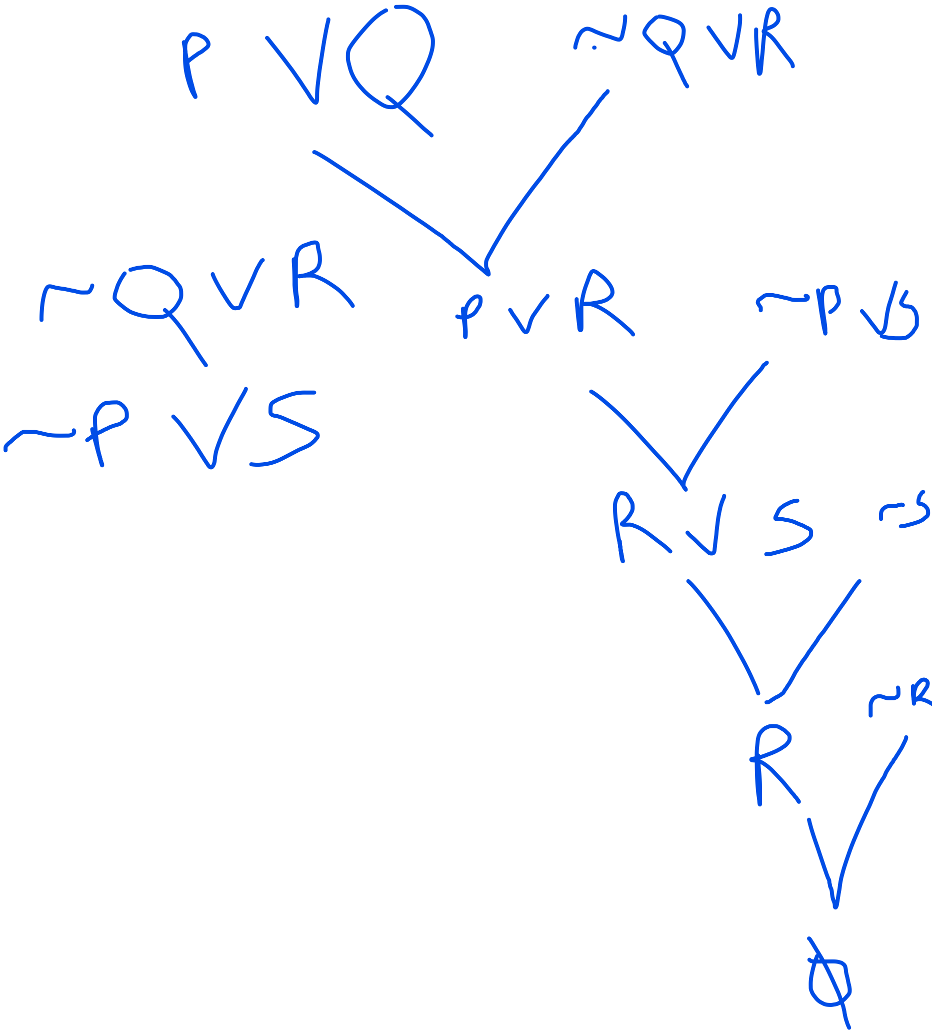
It will be represented as b**Q → R.** 2

3) P: Humidity is high. **...from(1)**

Let, S: It is hot.

It will be represented as **P → S.** 3

4) ¬S: It is not hot.



¬∃x: A becomes ∀x: ¬A

It means that the universal quantifier becomes existential quantifier and vice-versa**.**

- **Standardize variables:** If two sentences use same variable, it is required to change the name of one variable. This step is taken so as to remove the confusion when the quantifiers will be dropped.

For example: { ∀x: A(x) V ∀x: B(x)}

- **Skolemize:** It is the process of removing existential quantifier through elimination.
- **Drop universal quantifiers:** If we are on this step, it means all remaining variables must be universally quantified. Drop the quantifier.
- **Distribute V over Λ:** Here, the nested conjunction and disjunction are flattened.

Example of FOPL resolution

Consider the following knowledge base:

1. Gita likes all kinds of food.
2. Mango and chapati are food.
3. Gita eats almond and is still alive.
4. Anything eaten by anyone and is still alive is food.

Goal: Gita likes almond.

Solution: Convert the given sentences into FOPL as:

Let, x be the light sleeper.

1. ∀x: food(x) → likes(Gita,x)
2. food(Mango),food(chapati)
3. ∀x∀y: eats(x,y) Λ ¬ killed(x → food(y)
4. eats(Gita, almonds) Λ alive(Gita)
5. ∀x: ¬killed(x) → alive(x)
6. ∀x: alive(x) → ¬killed(x)

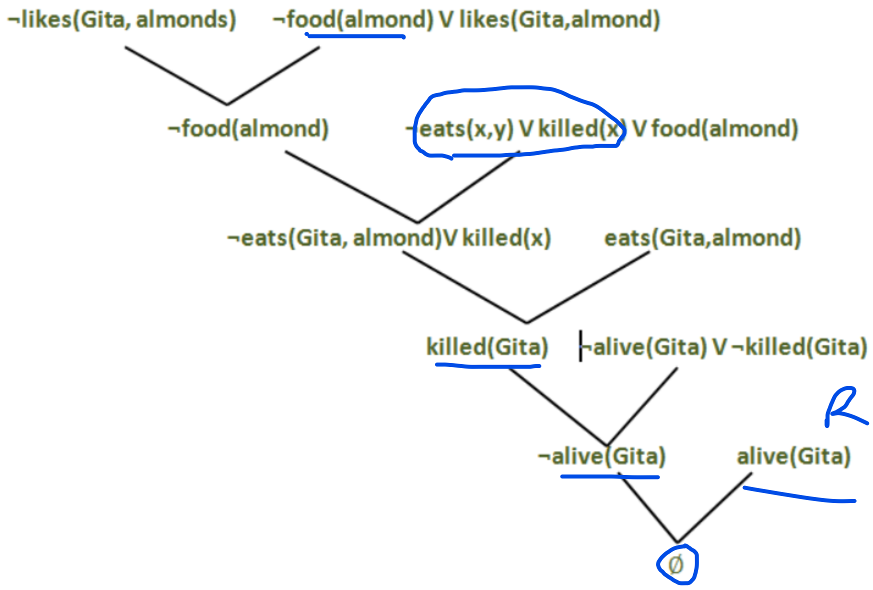
Goal: likes(Gita, almond)

Negated goal: ¬likes(Gita, almond)

Now, rewrite in CNF form:

1. ¬food(x) V likes(Gita, x)
2. food(Mango),food(chapati)
3. ¬eats(x,y) V killed(x) V food(y)
4. eats(Gita, almonds), alive(Gita)
5. killed(x) V alive(x)
6. ¬alive(x) V ¬killed(x)

Finally, construct the resolution graph:


¬likes(Gita, almonds)
¬food(almond) V likes(Gita,almond)
¬food(almond)
¬eats(x,y) V killed(x) V food(almond)
¬eats(Gita, almond)V killed(x)
eats(Gita,almond)
killed(Gita)
¬alive(Gita) V ¬killed(Gita)
R
¬alive(Gita)
alive(Gita)
Ø

Hence, we have achieved the given goal with the help of Proof by Contradiction. Thus, it is proved that Gita likes almond.