Unit III- KNOWLEDGE AND REASONING

Knowledge and Reasoning
Table of Contents

- Knowledge and reasoning-Approaches and issues of knowledge reasoning-Knowledge base agents
- Logic Basics-Logic-Propositional logic-syntax ,semantics and inferences-Propositional logic- Reasoning patterns
- Unification and Resolution-Knowledge representation using rules-Knowledge representation using semantic nets
- Knowledge representation using frames-Inferences-
- Uncertain Knowledge and reasoning-Methods-Bayesian probability and belief network
- Probabilistic reasoning-Probabilistic reasoning over time-Probabilistic reasoning over time
- Other uncertain techniques-Data mining-Fuzzy logic-Dempster -shafer theory

UNIFICATION

- Unification is a process of making two different logical atomic expressions identical by finding a substitution.

- Unification depends on the substitution process.

- E.g.: P(x, F(y)), P(a, F(g(z)))
- Unification: [a|x, g(z)|y]

- Substitute x with a & y with g(z); it will be represented as a|x & g(z)|y
- With the substitutions, the first expression will be identical to the second expression & the substitution set will be [a|x, g(z)|y]

UNIFICATION

- The UNIFY algorithm is used for unification, which takes two atomic sentences and returns a unifier for those sentences (If any exist).
- Unification is a key component of all first-order inference algorithms.
- It returns fail if the expressions do not match with each other.
- The substitution variables are called Most General Unifier or MGU.

UNIFICATION

Conditions for Unification:

Following are some basic conditions for unification:

- Predicate symbol must be same, atoms or expression with different predicate symbol can never be unified.
- Number of Arguments in both expressions must be identical.
- Unification will fail if there are two similar variables present in the same expression.

UNIFICATION

Algorithm: Unify(Ψ_1, Ψ_2)

```
Step. 1: If Ψ1 or Ψ2 is a variable or constant, then:
        a) If Ψ1 or Ψ2 are identical, then return NIL.
        b) Else if Ψ1is a variable,
                a. then if Ψ1 occurs in Ψ2, then return FAILURE
                b. Else return { (Ψ2/ Ψ1)}.
        c) Else if Ψ2 is a variable,
                a. If Ψ2 occurs in Ψ1 then return FAILURE,
                b. Else return {( Ψ1/ Ψ2)}.
        d) Else return FAILURE.
```

UNIFICATION

```
Step.2: If the initial Predicate symbol in Ψ1 and Ψ2 are not same, then return FAILURE.
Step. 3: IF Ψ1 and Ψ2 have a different number of arguments, then return FAILURE.
Step. 4: Set Substitution set(SUBST) to NIL.
Step. 5: For i=1 to the number of elements in Ψ1.
        a) Call Unify function with the ith element of Ψ1 and ith element of Ψ2, and put the result into S.
        b) If S = failure then returns Failure
        c) If S ≠ NIL then do,
                a. Apply S to the remainder of both L1 and L2.
                b. SUBST= APPEND(S, SUBST).
Step.6: Return SUBST.
```

https://www.javatpoint.com/ai-unification-in-first-order-logic

UNIFICATION

Consider P(x, g(x))

- (1) P(z, y): Unifies with [x|z, g(x)|y]
- (2) P(z, g(z)): Unifies with [x|z, z|x]
- (3) P(prime, f(prime)): Does not unify (g and f doesn't match)

https://www.javatpoint.com/ai-unification-in-first-order-logic

Resolution

Resolution

Resolution is a theorem proving technique that proceeds by building refutation proofs, i.e., proofs by contradictions. It was invented by a Mathematician John Alan Robinson in the year 1965.

Resolution is used, if there are various statements are given, and we need to prove a conclusion of those statements. Unification is a key concept in proofs by resolutions. Resolution is a single inference rule which can efficiently operate on the **conjunctive normal form or clausal form**.

Clause: Disjunction of literals (an atomic sentence) is called a **clause**. It is also known as a unit clause.

Conjunctive Normal Form: A sentence represented as a conjunction of clauses is said to be **conjunctive normal form** or **CNF**.

Resolution

Steps for Resolution:

1. Conversion of facts into first-order logic.
2. Convert FOL statements into CNF
3. Negate the statement which needs to prove (proof by contradiction)
4. Draw resolution graph (unification).

Resolution

For example we have following **statements**,

(1) If it is a pleasant day you will do strawberry picking

(2) If you are doing strawberry picking you are happy.

```
(1) strawberry_picking ← pleasant
(2) happy ← strawberry_picking
```

And again these statements can be written **in CNF** like this -

```
(1) (strawberry_picking V~pleasant) ∧
(2) (happy V~strawberry_picking)
```

By resolving these two clauses and cancelling out the conflicting terms 'strawberry_picking' and '~strawberry_picking', we can have one **new clause**,

```
(3) ~pleasant V happy
```

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.

Resolution

1. Let, P: Humidity is high.

   Q: Sky is cloudy.

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

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

Let, R: It will rain.

It will be represented as b**Q** ? **R.**

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

Let, S: It is hot.

It will be represented as **P** ? **S.**

4) ¬S: It is not hot.

Resolution

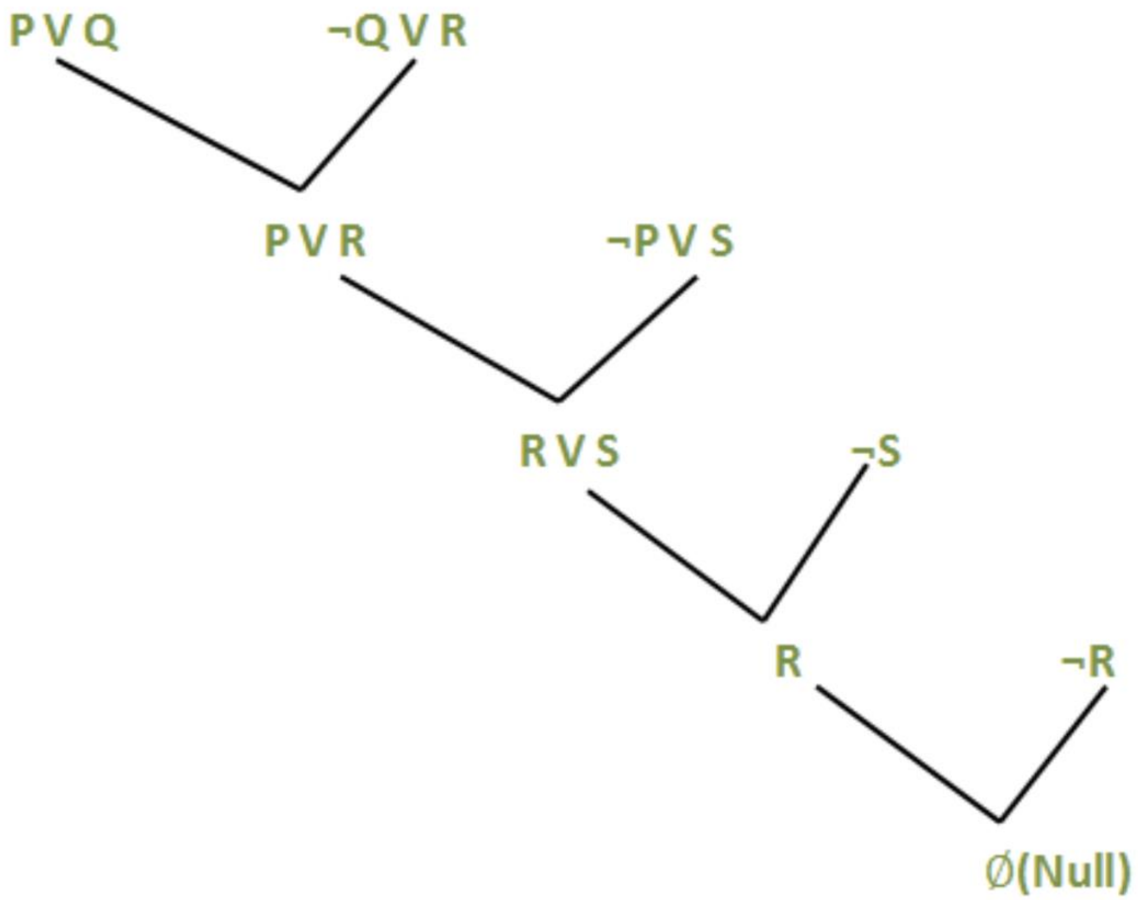
Applying resolution method:

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.

Resolution



Resolution

https://www.tutorialandexample.com/resolution-method-in-ai

Thank You