

<b>Date:</b> <b>Ex No:</b>  7.2	<b>Title of the Lab</b> Implementation of Resolution in SWI Prolog	<b>Name:</b> Mainak Chaudhuri <b>Registration Number:</b>  RA1911027010039 <b>Section:</b> N1 <b>Lab Batch:</b> 1 <b>Day Order:</b> 3
--	--	---

AIM:

To implement Resolution in SWI Prolog.

Description of the Concept or Problem given:

In simple words resolution is inference mechanism. Let's say we have clauses  $m :- b.$  and  $t :- p, m, z.$  So from that we can infer  $t :- p, b, z.$  - that is called resolution. Means, when you resolve two clauses you get one new clause. Another easy example, we have two sentences (1) All women like shopping. (2) Olivia is a woman. Now we ask query 'Who likes shopping'. So, by resolving above sentences we can have one new sentence Olivia likes shopping.

Manual Solution:

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).

Screenshots of the Outputs:

The screenshot shows the SWI-Prolog IDE with a file named 'hobbies.pl'. The program defines a set of facts and a rule for finding friends. The execution trace shows the results of the query `?- friends(ali, bob).` and `?- friends(ali, cal).`.

```

hobbies.pl
person(ali,20).
person(bob,20).
person(cal,25).

hobby(ali,skiing).
hobby(bob,skiing).
hobby(cal,skiing).

friends(P1,P2):-
  hobby(P1,H),
  hobby(P2,H),
  P1\=P2,
  person(P1,A1),
  person(P2,A2),
  AD is abs(A2-A1),
  AD<=3.

```

```

Welcome to SWI-Prolog (threaded, 64 bits, version 8.4.2)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- friends(ali,bob).
true.

?- trace.
true.

[trace] ?- friends(ali,bob).
Call: (10) friends(ali, bob) ? creep
Exit: (11) hobby(ali, _18582) ? creep
Exit: (11) hobby(ali, skiing) ? creep
Call: (11) hobby(bob, skiing) ? creep
Exit: (11) hobby(bob, skiing) ? creep
Call: (11) ali\=bob ? creep
Exit: (11) ali\=bob ? creep
Call: (11) person(ali, 20) ? creep
Exit: (11) person(ali, 20) ? creep
Call: (11) person(bob, 20) ? creep
Exit: (11) person(bob, 20) ? creep
Call: (11) 0 is abs(20-20) ? creep
Exit: (11) 0 is abs(20-20) ? creep
Call: (11) 0<=3 ? creep
Exit: (11) 0<=3 ? creep
Exit: (10) friends(ali, bob) ? creep
true.

[trace] ?- friends(ali,cal).
Call: (10) friends(ali, cal) ? creep
Exit: (11) hobby(ali, _32546) ? creep
Exit: (11) hobby(ali, skiing) ? creep
Call: (11) hobby(cal, skiing) ? creep
Exit: (11) hobby(cal, skiing) ? creep
Call: (11) ali\=cal ? creep
Exit: (11) ali\=cal ? creep
Call: (11) person(ali, 20) ? creep
Exit: (11) person(ali, 20) ? creep
Call: (11) person(cal, 25) ? creep
Exit: (11) person(cal, 25) ? creep
Call: (11) 5 is abs(25-20) ? creep
Exit: (11) 5 is abs(25-20) ? creep
Call: (11) 5<=3 ? creep
Fail: (11) 5<=3 ? creep
Fail: (10) friends(ali, cal) ? creep
false.

[trace] ?-

```

Signature of the Student

[MAINAK CHAUDHURI]