29/12/23

Date __/__/__
Page

LAB-7

KNOWLEDGE BASE RESOLUTION.

```
def negate_literal (literal):
    if literal[0] == '~':
        return literal[1:]
    else:
        return '~' + literal

def resolve (c1, c2):
    resolved_clause = set (c1) | set (c2)

    for literal in c1:
        if negate_literal (literal) in c2:
            resolved_clause.remove (literal)
            resolved_clause.remove (negate_
                                    literal (literal))
    return tuple (resolved_clause)

def resolution (knowledge_base):
    while True:
        new_clauses = set()
        for i, c1 in enumerate (knowledge_base):
            for j, c2 in enumerate (knowledge_base)
                if i != j:
                    new_clause = resolve (c1, c2)
                    if len (new_clause) > 0 and new_clau
                                    not in knowledge_ba
                        new_clauses.add (new_clause)

        if not new_clauses:
            break
        knowledge_base |= new_clauses
    return knowledge_base
```

Date __/__/__

Page _____

```
if __name__ == "__main__":
    kb = {('p','q'),('~p','r'),('~q','~r')}
    result = resolution(kb)
    print("Original KB", kb)
    print("Resolved KB", result)
```

2nd

Harshitha R-1BM21CS075

Step	Clause	Derivation
1.	Rv~P	Given.
2.	Rv~Q	Given.
3.	~RvP	Given.
4.	~RvQ	Given.
5.	~R	Negated conclusion.
6.		Resolved Rv~P and ~RvP to Rv~R, which is in turn null.

A contradiction is found when ~R is assumed as true. Hence, R is true.

Harshitha R-1BM21CS075

Step	Clause	Derivation
1.	PvQ	Given.
2.	~PvR	Given.
3.	~QvR	Given.
4.	~R	Negated conclusion.
5.	QvR	Resolved from PvQ and ~PvR.
6.	PvR	Resolved from PvQ and ~QvR.
7.	~P	Resolved from ~PvR and ~R.
8.	~Q	Resolved from ~QvR and ~R.
9.	Q	Resolved from ~R and QvR.
10.	P	Resolved from ~R and PvR.
11.	R	Resolved from QvR and ~Q.
12.		Resolved R and ~R to Rv~R, which is in turn null.

A contradiction is found when ~R is assumed as true. Hence, R is true.

Harshitha R-1BM21CS075

Step	Clause	Derivation
1.	PvQ	Given.
2.	PvR	Given.
3.	~PvR	Given.
4.	RvS	Given.
5.	Rv~Q	Given.
6.	~Sv~Q	Given.
7.	~R	Negated conclusion.
8.	QvR	Resolved from PvQ and ~PvR.
9.	Pv~S	Resolved from PvQ and ~Sv~Q.
10.	P	Resolved from PvR and ~R.
11.	~P	Resolved from ~PvR and ~R.
12.	Rv~S	Resolved from ~PvR and Pv~S.
13.	R	Resolved from ~PvR and P.
14.	S	Resolved from RvS and ~R.
15.	~Q	Resolved from Rv~Q and ~R.
16.	Q	Resolved from ~R and QvR.
17.	~S	Resolved from ~R and Rv~S.
18.		Resolved ~R and R to ~RvR, which is in turn null.

A contradiction is found when ~R is assumed as true. Hence, R is true.