Date / /
Page
SPLASH

29-12-2020 LAB-Test-2 AI

C. Karan Naidu
1BM18CS042

3. Given P => Q and R => S Prove PVR => QVS by resolution. Assume all the sentenseces are in CNF form.

```
def disjunctify(clauses):
    disjuncts = [ ]
for clauses in clauses:
disjuncts.append(tuple(clause.split('v')))
return disjuncts.

def getResolvant(ci, cj, di, dj):
resolvant = list(ci) + list(cj)
resolvant.remove(di)
resolvant.remove(dj)
return tuple(resolvant).

def resolve(ci, cj):
for di in ci:
for dj in cj:
    if di == '~'+dj or dj == '~'+di:
        return getResolvant(ci, cj, di, dj)

def checkResolution(clauses, query):
clauses += [query if query.startswith('~')
            else '~'+query]
proposition = '^'.join('('+clause+')' for
                clause in clauses)
print('Trying to prove', proposition, 'by
                contradiction...')
```

```
clauses = disjunctify(clauses)
resolved = False

new = set()

while not resolved:
n = len(clauses)
pairs = [(clauses[i], clauses[j])
  for i in range(n) for j in range(i+1, n)]

for (ci, cj) in pairs:
 resolvant = resolve(ci, cj)

 if not resolvant:
    resolved = True
    break

new = new.union(set(resolvents))

if new.issubset(set(clauses)):
    break
 for clause in new:
 if clause not clauses:
    clauses.append(clause)

if resolved:
    print("Knowledge base entails the query,
          proved by resolution")
else:
    print("Knowledge base doesn't entail the query
     no empty set produced after resolution")
```

Date __/__/__
Page
SPLASH

```
clauses = input ('Enter clauses').split()
query = input ('Enter the query:')
check Resolution (clauses, query)
```