Lab 5

Given in question

Code

from itertools import product

# Step 1: Define Knowledge Base (KB)

def KB(P, Q, R):

"""

Knowledge Base:

1. Q → P means: not Q or P

2. P → ¬Q means: not P or not Q

3. Q ∨ R

"""

return (not Q or P) and (not P or not Q) and (Q or R)

# Step 2: Define Queries

def query\_R(P, Q, R):

return R

def query\_R\_implies\_P(P, Q, R):

return (not R) or P # R → P

def query\_Q\_implies\_R(P, Q, R):

return (not Q) or R # Q → R

def query\_notR\_implies\_Q(P, Q, R):

return R or Q # ¬R → Q

queries = {

"R": query\_R,

"R → P": query\_R\_implies\_P,

"Q → R": query\_Q\_implies\_R,

"¬R → Q": query\_notR\_implies\_Q

}

# Step 3: Truth Table Entailment Function

def entails(KB\_func, query\_func):

print(f"\nChecking entailment for query: {query\_func.\_\_name\_\_}")

print("P\tQ\tR\tKB\_True\tQuery\_True")

all\_true = True

for P, Q, R in product([True, False], repeat=3):

KB\_true = KB\_func(P, Q, R)

query\_true = query\_func(P, Q, R)

print(f"{P}\t{Q}\t{R}\t{KB\_true}\t{query\_true}")

if KB\_true and not query\_true:

print(f"❌ Counterexample found at (P={P}, Q={Q}, R={R})")

all\_true = False

break

if all\_true:

print("✅ No counterexample found.")

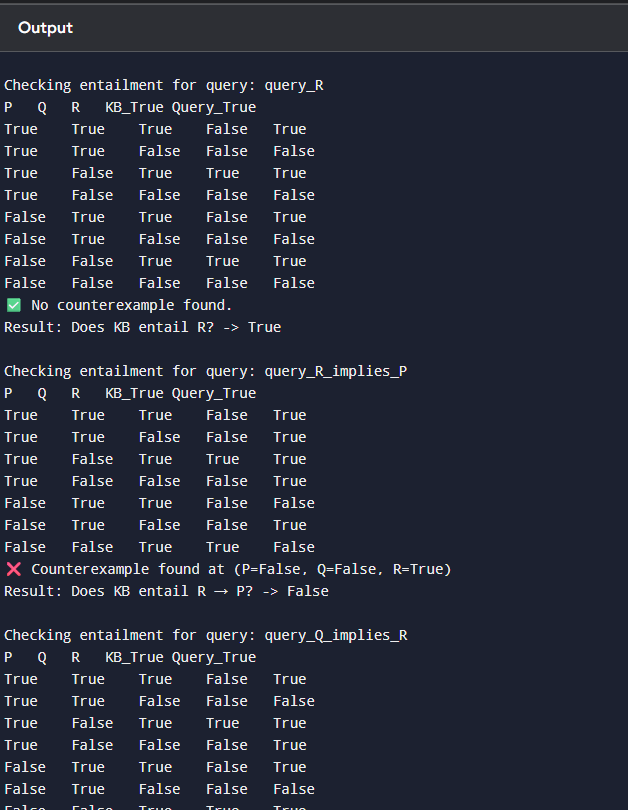
return all\_true

# Step 4: Run for each query

for name, func in queries.items():

result = entails(KB, func)

print(f"Result: Does KB entail {name}? -> {result}")

OUTPUT:  


EXAMPLE IN PDF

from itertools import product

# Define individual formula parts

def A\_or\_C(A, B, C):

return A or C

def B\_or\_notC(A, B, C):

return B or (not C)

def KB(A, B, C):

# Knowledge Base: (A ∨ C) ∧ (B ∨ ¬C)

return (A\_or\_C(A, B, C)) and (B\_or\_notC(A, B, C))

def alpha(A, B, C):

# Query: A ∨ B

return A or B

# Truth table and entailment check

def entails(KB\_func, alpha\_func):

print("A\tB\tC\tA∨C\tB∨¬C\tKB\tα")

entailment = True # assume KB ⊨ α unless we find counterexample

for A, B, C in product([False, True], repeat=3):

a\_or\_c = A\_or\_C(A, B, C)

b\_or\_notc = B\_or\_notC(A, B, C)

kb\_val = KB\_func(A, B, C)

alpha\_val = alpha\_func(A, B, C)

# Print truth table row

print(f"{A}\t{B}\t{C}\t{a\_or\_c}\t{b\_or\_notc}\t{kb\_val}\t{alpha\_val}")

# If KB true and α false → entailment fails

if kb\_val and not alpha\_val:

entailment = False

print("\nChecking entailment: Does KB ⊨ α ?")

if entailment:

print("✅ Yes, KB entails α (no counterexamples found).")

else:

print("❌ No, KB does NOT entail α (found a counterexample).")

# Run the truth table entailment check

entails(KB, alpha)

OUTPUT

