**Week 5 – Propositional logic**

from itertools import product

# Define propositional logic operations

def implies(a, b):

return (not a) or b

# Knowledge Base sentences

def KB(P, Q, R):

s1 = implies(Q, P) # Q → P

s2 = implies(P, not Q) # P → ¬Q

s3 = Q or R # Q ∨ R

return s1 and s2 and s3 # KB is true only if all hold

# All combinations of truth values for P, Q, R

values = list(product([False, True], repeat=3))

print("P\tQ\tR\tQ→P\tP→¬Q\tQ∨R\tKB")

print("-"\*50)

models = []

for P, Q, R in values:

s1 = implies(Q, P)

s2 = implies(P, not Q)

s3 = Q or R

kb\_val = s1 and s2 and s3

print(f"{P}\t{Q}\t{R}\t{s1}\t{s2}\t{s3}\t{kb\_val}")

if kb\_val:

models.append((P, Q, R))

print("\n Models where KB is True:", models)

# Check entailments

entails\_R = all(R for P, Q, R in models)

entails\_R\_imp\_P = all((not R) or P for P, Q, R in models)

entails\_Q\_imp\_R = all((not Q) or R for P, Q, R in models)

print("\nEntailments:")

print("KB ⊨ R :", entails\_R)

print("KB ⊨ R → P :", entails\_R\_imp\_P)

print("KB ⊨ Q → R :", entails\_Q\_imp\_R)

Output:

