Lab 6
Propositional Logic

[1]:
```
from itertools import product
def implies(a, b):
    return (not a) or b
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
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)
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)
```

```
print( KB ⊨ Q → R : , entails_Q_imp_R)
```

```
P       Q       R       Q→P     P→¬Q    Q∨R     KB
--------------------------------------------------
False   False   False   True    True    False   False
False   False   True    True    True    True    True
False   True    False   False   True    True    False
False   True    True    False   True    True    False
True    False   False   True    True    False   False
True    False   True    True    True    True    True
True    True    False   True    False   True    False
True    True    True    True    False   True    False

 Models where KB is True: [(False, False, True), (True, False, True)]

Entailments:
KB ⊨ R : True
KB ⊨ R → P : False
KB ⊨ Q → R : True
```

Example Question

## Propositional Inference: Enumeration Method

**Example**

$$\alpha = A \vee B \qquad KB = (A \vee C) \wedge (B \vee \neg C)$$

**Checking that $KB \models \alpha$**

| $A$ | $B$ | $C$ | $A \vee C$ | $B \vee \neg C$ | $KB$ | $\alpha$ |
|---|---|---|---|---|---|---|
| *false* | *false* | *false* | *false* | *true* | *false* | *false* |
| *false* | *false* | *true* | *true* | *false* | *false* | *false* |
| *false* | *true* | *false* | *false* | *true* | *false* | *true* |
| *false* | *true* | *true* | *true* | *true* | *true* | *true* |
| *true* | *false* | *false* | *true* | *true* | *true* | *true* |
| *true* | *false* | *true* | *true* | *false* | *false* | *true* |
| *true* | *true* | *false* | *true* | *true* | *true* | *true* |
| *true* | *true* | *true* | *true* | *true* | *true* | *true* |

•[2]:
```python
from itertools import product
def implies(a, b):
    return (not a) or b
def KB(A, B, C):
    s1 = A or C  # A ∨ C
    s2 = B or (not C)  # B ∨ ¬C
    return s1 and s2  # KB is true only if both hold
def alpha(A, B):
    return A or B  # A ∨ B
values = list(product([False, True], repeat=3))
print("A\tB\tC\tA∨C\tB∨¬C\tKB\tα")
print("-"*50)
models = []
for A, B, C in values:
    s1 = A or C
    s2 = B or (not C)
    kb_val = s1 and s2
    alpha_val = A or B
    print(f"{A}\t{B}\t{C}\t{s1}\t{s2}\t{kb_val}\t{alpha_val}")
    if kb_val:
        models.append((A, B, C))
entails_alpha = all(alpha(A, B) for A, B, C in models)
print("\nEntailments:")
print("KB ⊨ α :", entails_alpha)
```

```
    if kb_val:
        models.append((A, B, C))
entails_alpha = all(alpha(A, B) for A, B, C in models)
print("\nEntailments:")
print("KB ⊨ α :", entails_alpha)
```

```
A       B       C       A∨C     B∨¬C    KB      α
--------------------------------------------------
False   False   False   False   True    False   False
False   False   True    True    False   False   False
False   True    False   False   True    False   True
False   True    True    True    True    True    True
True    False   False   True    True    True    True
True    False   True    True    False   False   True
True    True    False   True    True    True    True
True    True    True    True    True    True    True

Entailments:
KB ⊨ α : True
```