Lab - 6

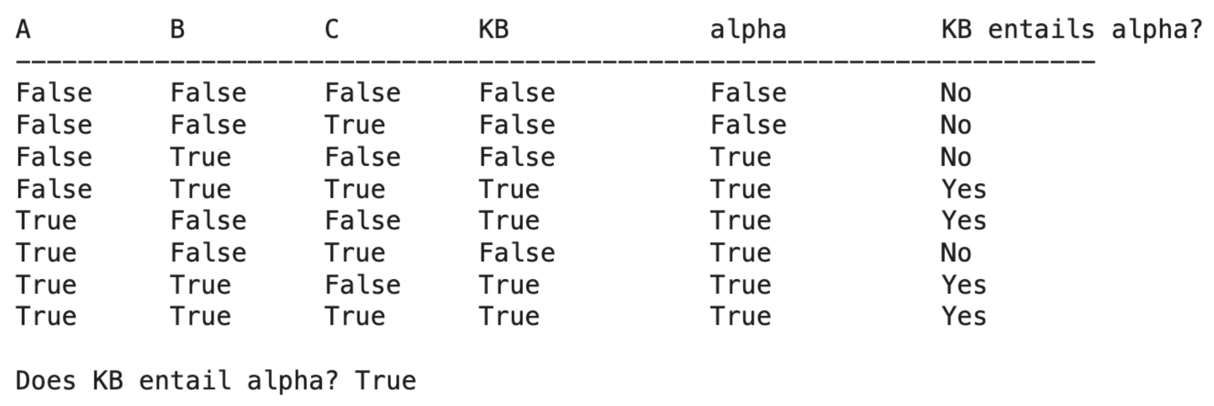
Propositional Logic

```python
import itertools

def evaluate_formula(formula, valuation):
    """
    Evaluate the propositional formula under the given 
truth assignment (valuation).
    The formula is a string of logical operators like 
'AND', 'OR', 'NOT', and can contain variables 'A', 'B', 
'C'.
    """
    # Create a local environment (dictionary) for 
variable assignments
    env = {var: valuation[i] for i, var in 
enumerate(['A', 'B', 'C'])}

    # Replace logical operators with Python equivalents
    formula = formula.replace('AND', 'and').replace('OR', 
'or').replace('NOT', 'not')

    # Replace variables in the formula with their 
corresponding truth values
    for var in env:
        formula = formula.replace(var, str(env[var]))

    # Evaluate the formula and return the result (True or 
False)
    try:
        return eval(formula)
    except Exception as e:
        raise ValueError(f"Error in evaluating formula: 
{e}")

def truth_table(variables):
    """
    Generate all possible truth assignments for the given 
variables.
    """
    return list(itertools.product([False, True], 
repeat=len(variables)))
```

```python
def entails(KB, alpha):
    """
    Decide if KB entails alpha using a truth-table 
enumeration algorithm.
    KB is a propositional formula (string), and alpha is 
another propositional formula (string).
    """
    # Generate all possible truth assignments for A, B, 
and C
    assignments = truth_table(['A', 'B', 'C'])

    print(f"{'A':<10}{'B':<10}{'C':<10}{'KB':<15}
{'alpha':<15}{'KB entails alpha?'}")  # Header for the 
truth table
    print("-" * 70)  # Separator for readability

    for assignment in assignments:
        # Evaluate KB and alpha under the current 
assignment
        KB_value = evaluate_formula(KB, assignment)
        alpha_value = evaluate_formula(alpha, assignment)

        # Print the current truth assignment and the 
results for KB and alpha
        print(f"{str(assignment[0]):<10}
{str(assignment[1]):<10}{str(assignment[2]):<10}
{str(KB_value):<15}{str(alpha_value):<15}{'Yes' if 
KB_value and alpha_value else 'No'}")

        # If KB is true and alpha is false, then KB does 
not entail alpha
        if KB_value and not alpha_value:
            return False

    # If no counterexample was found, then KB entails 
alpha
    return True

# Define the formulas for KB and alpha
alpha = 'A OR B'
KB = '(A OR C) AND (B OR NOT C)'
```

```python
# Check if KB entails alpha
result = entails(KB, alpha)

# Print the final result of entailment
print(f"\nDoes KB entail alpha? {result}")
```

Output:



```
A          B          C          KB              alpha           KB entails alpha?
----------------------------------------------------------------------
False      False      False      False           False           No
False      False      True       False           False           No
False      True       False      False           True            No
False      True       True       True            True            Yes
True       False      False      True            True            Yes
True       False      True       False           True            No
True       True       False      True            True            Yes
True       True       True       True            True            Yes

Does KB entail alpha? True
```