Lab 4 Dylan Liesenfelt

June 7, 2024

1 1. Logical functions and laws using conditionals

TASK: Read and modify code acordingly

[ ]:
```python
# Logical functions
def implies(p, q):
    return not p or q

def contrapositive(p, q):
    return implies(not q, not p)

def converse(p, q):
    return implies(q, p)

def inverse(p, q):
    return implies(not p, not q)

def biconditional(p, q):
    return p == q

def xor(p,q):
    return (p or q) and (not (p and q))

# Logical equivalences
def distributive(p, q, r):
    return implies(p or q, r) == (implies(p, r) and implies(q, r))

def exportation(p, q, r):
    return implies(p and q, r) == implies(p, implies(q, r))

def reduction(p, q):
    return not implies(p, q) == (p and not q)

def equivalence(p,q):
    return biconditional(p,q) == (not xor(p,q))

def truthTable():
    # Columns
```

```
    # TASK: Modify each column as required for your compound statement
    # p, q, r, ¬p   q, (p → r) v (q → r)
    col1 = 'p'
    col2 = 'q'
    col3 = 'r'
    col4 = '¬p   q'
    col5 = '(p   q) v (q → r)'
    print(f'{col1} \t{col2} \t{col3} \t{col4} \t{col5}')
    print('-'*45)

    # Iteration of proposition values
    for p in [True, False]:
        for q in [True, False]:
            for r in [True, False]:
                # Columns
                # TASK: Modify each column as required for your compound␣
 ↪statement
                col1 = p
                col2 = q
                col3 = r
                col4 = (not p) and q
                col5 = implies(p, r) or implies(q,r)
                print(f'{col1} \t{col2} \t{col3} \t{col4} \t{col5}')

truthTable()
```

```
p       q       r       ¬p   q   (p   q) v (q → r)
---------------------------------------------
True    True    True    False   True
True    True    False   False   False
True    False   True    False   True
True    False   False   False   True
False   True    True    True    True
False   True    False   True    True
False   False   True    False   True
False   False   False   False   True
```

1.1 2. Testing each logic function

[ ]:
```
# Task: Modify values for p and q
#       and try each logical function and logical equivalence
# Example:
p = True
q = False

# Example:
print( (not xor(p,q)) )
```

```
print( biconditional(p,q) )
print(xor(p,q))
print(inverse(p,q))
print(converse(p,q))
```

```
False
False
True
True
True
```

1.2 3. Compound statements using conditionals in English

[ ]:
```
def proposition_to_string(p, q):
    prop = f"If {p}, then {q}."
    return prop

# TASK: Replace p and q for another compound statement
def main():
    p = "it is hot"
    q = "people go to the beach"

    print("Original Proposition: ", proposition_to_string(p, q))

    p_inv = "it is cold"
    q_inv = "people do not go to the beach"

    print("Inverse: ", proposition_to_string(p_inv, q_inv))
    print("Converse: ", proposition_to_string(q, p))
    print("Contrapositive: ", proposition_to_string(q_inv, p_inv))

main()
```

```
Original Proposition:  If it is hot, then people go to the beach.
Inverse:  If it is cold, then people do not go to the beach.
Converse:  If people go to the beach, then it is hot.
Contrapositive:  If people do not go to the beach, then it is cold.
```

1.3 4. Solve some of the true tables on step 1 by hand

[ ]:
```
```