Assignment 4

February 22, 2020

1 Assignment 4

1.1 Problem 1

Create a function fa(p,q,r). It returns the value of the logical expression

$$p \wedge (q \vee r)$$

Use this function to print a complete truth table for the logical expression.

[5]:
```
tValues = [True, False]

def fa(p,q,r):
    return (p and (q or r))

for p in tValues:
    for q in tValues:
        for r in tValues:
                print(p,q,r, fa(p,q,r))
```

```
True True True True
True True False True
True False True True
True False False False
False True True False
False True False False
False False True False
False False False False
```

1.2 Problem 2

Create a function fb(p,q,r). It returns the value of the logical expression

$$(p \wedge q) \vee (p \wedge r))$$

Use this function to print a complete truth table for the logical expression.

[22]:
```
def fb(p,q,r):
    return ((p and q) or (p and r))

for p in tValues:
    for q in tValues:
        for r in tValues:
                print(p,q,r, fb(p,q,r))
```

```
True True True True
True True False True
True False True True
True False False False
False True True False
False True False False
False False True False
False False False False
```

1.3 Problem 3

Write a python program which uses the two functions you created to answer a question. Are these two logical expressions equivalent? You should modify the function compare(), which I demonstrated in class, to handle three logical

[26]:
```
def compare(fa,fb):

    # Initialize empty lists
    aList = []
    bList = []

    # Loop through possible values of p, q, r
    for p in tValues:
        for q in tValues:
            for r in tValues:
                A = fa(p,q,r)
                B = fb(p,q,r)

            # Append truth values for current p, q, r values
                aList.append(A)
                bList.append(B)

    # Determine conclusion
    if aList == bList:
        conclusion = "Yes - Equivalent"
    else:
        conclusion = "No - Not Equivalent"
    return conclusion
```

```
print(compare(fa,fb))
```

```
Yes - Equivalent
```

[0]: