Truth Tables
Problem 1

Compute the truth table for the following expressions and their subexpressions.

a) (p → q) ≡ (NOT p OR q)

p	q	p → q	NOT p	NOT p OR q	(p → q) = (NOT p OR q)
0	0	1	1	1	1
0	1	1	1	1	1
1	0	0	0	0	1
1	1	1	0	1	1

b) p → (q → (r OR NOT p))

p	q	r	NOT p	r OR NOT p	q → (r OR NOT p)	p → (q → (r OR NOT p))
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	1	1	1
1	0	0	0	0	1	1
1	0	1	0	1	1	1
1	1	0	0	0	0	0
1	1	1	0	1	1	1

c) (p OR q) → (p AND q)

p	q	p AND q	p OR q	(p AND q) → (p OR q)
0	0	0	0	1
0	1	0	1	1
1	0	0	1	1
1	1	1	1	1

Problem 2

The binary exclusive–or function, ⊗, is defined to have value TRUE if and only if exactly one of its arguments are TRUE.

a) Draw the truth table for ⊗.

p	**q**	**p ⊗ q**
0	0	0
0	1	1
1	0	1
1	1	0

b) Is ⊗ commutative? Is it associative?

I know ⊗is identitive, because in encryption we can use a randomly generated stream of binary bits and ⊗it with plain text to get ciphertext which is reversible to decrypt.

I'd say it's commutative since "**a ⊗ b = b ⊗ a**", I have no idea how to prove it though. I'd say associative as well since I'm 99% sure "**a ⊗ (b ⊗ c)= (a ⊗ b) ⊗ c**"

From Boolean Functions to Logical Expressions

Problem 3

The truth table below defines two Boolean functions, a and b, in terms of variables p, q, and r. Write sum-of-products expressions for each of these functions.

p	q	r	a	b
0	0	0	0	1
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	0
1	0	1	1	0
1	1	0	1	0
1	1	1	1	0

a: pqr + $\overline{pq}$r + pqr + pqr + p$\overline{q}$r + p$\overline{q}$r + pq$\overline{r}$ + pqr

b: $\overline{pqr}$ + $\overline{pq}$r + $\overline{p}$q$\overline{r}$ + pqr + pqr + pqr + pqr + $\overline{pqr}$

Problem 4

Write product-of-sums expressions for

a) Function a of the previous table.

   i) (p + q + r)(p + q + r)(p + !q + r)(p + !q + !r)(p + q + r)(p + q + r)(p + q + r)(p + q + r)

b) Function b of the previous table.

   i) (!p + !q + !r)(p + q + r)(p + q + r)(p + !q + !r)(!p + q + r)(!p + q + !r)(!p + !q + r)(p + q + r)

c) Function z of the table below.

   i) (x + y + c)(x + y + c)(x + y + c)(x + !y + !c)(x + y +c)(!x + y + !c)(!x + !y + c)(x + q + y)

Designing Logical Expressions by Karnaugh Maps

Problem 5

Draw the Karnaugh maps for the following functions of variables p, q, r, and s.

a) The function that is TRUE if one, two, or three of p, q, r, and s are TRUE, but not if zero or all four are TRUE.

	pq				
rs		00	01	11	10
	00	0	1	1	1
	01	1	1	1	1
	11	1	1	0	1
	10	1	1	1	1

b) The function that is TRUE if up to two of p, q, r, and s are TRUE, but not if three or four are TRUE.

	rs				
pq		00	01	11	10
	00	1	1	1	1
	01	1	1	0	1
	11	1	0	0	0
	10	1	1	0	1

c) The function that is TRUE if one, three, or four of p, q, r, and s are TRUE, but not if zero or two are TRUE.

	pq				
rs		00	01	11	10
	00	0	1	0	1
	01	1	0	1	0
	11	0	1	1	1
	10	1	0	1	0

d) The function represented by the logical expression pqr → s.

	pq				
rs		00	01	11	10
	00	0	1	0	1
	01	1	0	1	0
	11	0	1	1	1
	10	1	0	1	0

e) The function that is TRUE if pqrs, regarded as a binary number, has value less than ten.

	pq				
rs		00	01	11	10
	00	1	1	0	1
	01	1	1	0	1
	11	1	1	0	0
	10	1	1	0	0

Problem 6

Find the implicants - other than the minterms - for each of your Karnaugh maps from Problem 5. Which of them are prime implicants? For each function, find a sum of prime implicants that covers all the 1's of the map. Do you need to use all the prime implicants?

a) **!pr + p!r + !rs + !pq + r!s + p!q**
b) **!p!r + !p!q + !r!s + !q!r + !p!s+ !q!s**
   i) Conjunctive

1) (!q + !r + !s)
2) (!p + !q + !s)
3) (!p + !q + !r)
4) (!p + !r + !s)

c) **pqs + pqr + prs + qrs**
d) **pqs + pqr + prs + qrs**
e) **!p + !r**

Tautologies

Problem 7

Which of the following expressions are tautologies?

a) pqr → p + q **YES**

p	q	r	pqr	p + q	pqr → p + q
0	0	0	0	0	1
0	0	1	0	0	1
0	1	0	0	1	1
0	1	1	0	1	1
1	0	0	0	1	1
1	0	1	0	1	1
1	1	0	0	1	1
1	1	1	1	1	1

b) ((p → q)(q → r)) → (p → r) **YES**

p	q	r	p → r	q → r	(p → r)(q → r)	(p → r)(q → r) → (p → r)
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	1	0	0	1
0	1	1	1	1	1	1
1	0	0	0	1	0	1
1	0	1	1	1	1	1
1	1	0	0	0	0	1
1	1	1	1	1	1	1

c) (p → q) → p **NO**

p	**q**	**p → q**	**(p → q) → p**
0	0	1	0
0	1	1	0
1	0	0	1
1	1	1	1

d) (p ≡ (q + r)) → (q → pr) **YES**

p	**q**	**r**	**q + r**	**p = (q + r)**	**pr**	**q → pr**	**(p ≡ (q + r)) → (q → pr)**
0	0	0	0	1	0	1	1
0	0	1	1	0	0	1	1
0	1	0	1	0	0	1	1
0	1	1	1	0	0	1	1
1	0	0	0	0	0	1	1
1	0	1	1	1	1	1	1
1	1	0	1	1	0	1	1
1	1	1	1	1	1	1	1

Problem 8

Apply DeMorgan's laws to turn the following expressions into expressions where the only NOT's are applied to propositional variables (i.e., the NOT's appear in literals only).

1. NOT (pq + pr)
   a. NOT((pq)(NOT pr))
   b. NOT pq + NOT pr
   c. NOT p + NOT q + NOT NOT p + NOT NOT r
   d. **NOT p + NOT q + p + r**
2. NOT (NOT p + q(NOT (r + s)))
   a. NOT((NOT p) q(NOT(r + NOT s)))
   b. NOT NOT p + NOT q + NOT(NOT(r + NOT s))
   c. p + NOT q + NOT(r NOT s)
   d. p + NOT q + NOT r + NOT NOT s

e. **p + NOT q + NOT r + s**