Homework 2

Aiden Allen

Exercise 1

1. Using truth tables, prove the Propositional Calculus identities of Section 2.2.

The propositional calculus identities of section 2.2 state that:

- P ∧ ¬P ≡ false

T	T	F
T	F	F
F	T	F
F	F	F

As we can see in the truth table above this proposition can be concluded to be true, because the negation of P and P will always be false. This is because the negation makes T into F and vice versa, it is therefore impossible for the negation of P to equal P.

- P ∨ false ≡ P

T	F	T
F	F	F

As detailed in the truth table above T or False will always be T. We can further prove this through observing the truth table for the or statement.

T	T	T
T	F	T
F	T	T
F	F	F

T will always be equal to the result of P ∨ false ≡ P. This is because T or Q will always be true unless both T and Q are false.

- ¬(¬P) ≡ P

P	(¬P)	¬(¬P)
T	F	T
F	T	F

P will always be equal to the double negation of P. This is because P can either be true or false, when negated true because false, as well as false becomes true. Therefore we can conclude when negating a proposition twice it will be equal to the initial value.

- P → Q ≡ ¬P ∨ Q

P	Q	P → Q	¬P ∨ Q	- P → Q ≡ ¬P ∨ Q
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

As we can see in the truth table above the values are the same for both P → Q and ¬P ∨ Q. This is because P or Q encompasses every value but F and F. When a negation is applied to P however it is logically equivalent to p implying q.

- Contra-positive law: P → Q ≡ ¬Q → ¬P

P	Q	P -> Q	¬Q → ¬P	P → Q ≡ ¬Q → ¬P
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

The contra-positive law states that P implies Q is logically equivalent to not Q implies not P. This holds true due to not p implies q being the contra positive of p implies q.

- De Morgan's Law 1: ¬(P ∨ Q) ≡ ¬P ∧ ¬Q

P	Q	(P ∨ Q)	¬(P ∨ Q)	P ∧ Q	¬P ∧ ¬Q
T	T	T	F	T	F
T	F	T	F	F	F
F	T	T	F	F	F
F	F	F	T	F	T

As we can see through the truth table above the truth values are identical for ¬(P ∨ Q) and ¬P ∧ ¬Q. We can understand this to be true because the negation of an or statement will only be true when values of P and Q are both false.

- De Morgan's Law 2: ¬(P ∧ Q) ≡ ¬P ∨ ¬Q

P	Q	P ∧ Q	P ∨ Q	¬(P ∧ Q)	¬P ∨ ¬Q
T	T	T	T	F	F

P	Q	P ∧ Q	P ∨ Q	¬(P ∧ Q)	¬P ∨ ¬Q
T	F	F	T	T	T
F	T	F	T	T	T
F	F	F	F	T	T

We can see through the truth table above that this proposition is indeed true. We can verify this claim through informally observing the logic used throughout this proposition. For values of P and Q, the statement P and Q will only hold true for pT qT. For P or Q, the proposition will only be false when pF qF. However if we negate the entirety of P and Q then pT qT will be false. If we negate P or negate Q, then we are left with the very same values. This proves their logical equivalence.

- Commutative Law 1: P ∧ Q ≡ Q ∧ P

P	Q	P ∧ Q	Q ∧ P	P ∧ Q ≡ Q ∧ P
T	T	T	T	T
T	F	F	T	T
F	T	F	T	T
F	F	F	F	T

As we can see regardless of the order of operations P and Q will always have the same result of Q and P.

- Commutative Law 2: P ∨ Q ≡ Q ∨ P

P	Q	P ∨ Q	Q ∨ P	P ∨ Q ≡ Q ∨ P
T	T	T	F	T
T	F	T	T	T
F	T	T	T	T
F	F	F	F	T

In much the same way as the and operation the or operation will also hold logical equivalence no matter the order of operations.

- Associative Law 1: (P ∧ Q) ∧ R ≡ P ∧ (Q ∧ R)

P	Q	R	(P ∧ Q)	(P ∧ Q) ∧ R	(Q ∧ R)	P ∧ (Q ∧ R)	(P ∧ Q) ∧ R ≡ P ∧ (Q ∧ R)
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	F	F	F	T
T	F	F	F	F	F	F	T
F	T	T	F	F	T	F	T
F	T	F	F	F	F	F	T
F	F	T	F	F	F	F	T
F	F	F	F	F	F	F	T

As we can see once again the order of operations does not matter in this case. This is because the same operation is being enacted on all of the axioms. This allows us to state that the truth table definitively proves their equivalence.

- Associative Law 2: (P ∨ Q) ∨ R ≡ P ∨ (Q ∨ R)

P	Q	R	P ∨ Q	Q ∨ R	(P ∨ Q) ∨ R	P ∨ (Q ∨ R)	(P ∨ Q) ∨ R ≡ P ∨ (Q ∨ R)
T	T	T	T	T	T	T	T
T	T	F	T	T	T	T	T
T	F	T	T	T	T	T	T
T	F	F	T	F	T	T	T
F	T	T	T	T	T	T	T
F	T	F	T	T	T	T	T
F	F	T	F	T	T	T	T
F	F	F	F	F	F	F	T

As we can see from the truth table above, it is obvious that the two statements are equivalent. We can establish this because the order of operations does not matter in this case.

- Distributive Law 1: P ∨ (Q ∧ R) ≡ (P ∨ Q) ∧ (P ∨ R)

P	Q	R	Q ∧ R	P ∨ (Q ∧ R)	(P ∨ Q)	(P ∨ R)	(P ∨ Q) ∧ (P ∨ R)	P ∨ (Q ∧ R) ≡ (P ∨ Q) ∧ (P ∨ R)
T	T	T	T	T	T	T	T	T
T	T	F	F	T	T	T	T	T
T	F	T	F	T	T	T	T	T
T	F	F	F	T	T	T	T	T
F	T	T	T	T	T	T	T	T
F	T	F	F	F	T	F	F	T
F	F	T	F	F	F	T	F	T
F	F	F	F	F	F	F	F	T

- Distributive Law 2: P ∧ (Q ∨ R) ≡ (P ∧ Q) ∨ (P ∧ R)

P	Q	R	(Q ∨ R)	P ∧ (Q ∨ R)	(P ∧ Q)	(P ∧ R)	(P ∧ Q) ∨ (P ∧ R)	P ∧ (Q ∨ R) ≡ (P ∧ Q) ∨ (P ∧ R)
T	T	T	T	T	T	T	T	T
T	T	F	T	T	T	F	T	T
T	F	T	T	T	F	T	T	T
T	F	F	F	F	F	F	F	T
F	T	T	T	F	F	F	F	T

P	Q	R	(Q ∨ R)	P ∧ (Q ∨ R)	(P ∧ Q)	(P ∧ R)	(P ∧ Q) ∨ (P ∧ R)	P ∧ (Q ∨ R) ≡ (P ∧ Q) ∨ (P ∧ R)
F	T	F	T	F	F	F	F	T
F	F	T	T	F	F	F	F	T
F	F	F	F	F	F	F	F	T

Exercise 2

2. A new operator, ⊕, or exclusive-or, may be defined by the following truth table (Figure 8). Create a propositional calculus expression using only ¬, ∨, and ∧ that is equivalent to P ⊕Q. Prove their equivalence using truth tables.

I propose that the expression P ⊕ Q is equivalent to (¬P ∧ Q) ∨ (¬Q ∧ P). As we can observe through the truth table below the two statements are equivalent. This is because the negation of both P and Q ensures that both P and Q cannot be true in order for the entire proposition to resolve in True.

P	Q		(¬P ∧ Q) ∨ (¬Q ∧ P)
T	T	F	F
T	F	T	T
F	T	T	T
F	F	F	F

Exercise 3

2. The logical operator "↔" is read "if and only if" (similarly as ≡. P ↔ Q is defined as being equivalent to (P → Q) ∧ (Q → P). Based on this definition, show that P ↔ Q is logically equivalent to (P ∨ Q) → (P ∧ Q):

Prove with Truth Tables

P	Q	(P ∨ Q)	(P ∧ Q)	(P ∨ Q) → (P ∧ Q)	P ↔ Q
T	T	T	T	T	T
T	F	T	F	F	F
F	T	T	F	F	F
F	F	F	F	T	T

Prove with substitution

P <--> Q is equivalent to (P -> Q) ∧ (Q -> P)

P -> Q is equivalent to ¬P ∨ Q

(P→Q)∧(Q→P) is equivalent to (¬P∨Q)∧(¬Q∨P)

¬P ∨ Q and ¬Q ∨ P

(P → Q) ∧ (Q → P) ≡ (¬P ∨ Q) ∧ (¬Q ∨ P)

(P ∨ Q) → (P ∧ Q) ≡ ¬(P ∨ Q) ∨ (P ∧ Q)

¬(P ∨ Q) ≡ ¬P ∧ ¬Q

¬(P ∨ Q) ∨ (P ∧ Q) ≡ (¬P ∧ ¬Q) ∨ (P ∧ Q)

(¬P ∧ ¬Q) ∨ (P ∧ Q) ≡ ((¬P ∨ (P ∧ Q)) ∧ (¬Q ∨ (P∧Q)))

¬Q ∨ (P ∧ Q) ≡ (¬Q ∨ P) ∧ (¬Q ∨ Q) ≡ (¬Q ∨ P) ∧ True≡ ¬Q∨P

(¬P ∨ Q) ∧ (¬Q ∨ P)

Exercise 4

4. Prove that implication is transitive in the propositional calculus, that is, that ((P → Q) ∧ (Q → R)) → (P → R)

P	Q	R	(P → Q)	(Q → R))	((P → Q) ∧ (Q → R))	(P → R)	((P → Q) ∧ (Q → R)) → (P → R)
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	F
T	F	T	F	T	F	T	F
T	F	F	F	T	F	F	F
F	T	T	T	T	T	T	T
F	T	F	T	F	F	T	F
F	F	T	T	T	T	T	T
F	F	F	T	F	F	T	F

P→Q is equivalent to ¬P ∨ Q

(P → Q) ∧ (Q → R) can be rewritten as: (¬P ∨ Q) ∧ (¬Q ∨ R)

¬P∨Q simplifies to Q (since ¬P is false).

Exercise 5

- Prove that Modus Ponens is sound for propositional calculus. Hint: use truth tables to enumerate all possible interpretations.

Modus Ponnes proposes that if P implies Q, and P is set to be true than Q must also be true.

We can see this through a real world example of decision making. We can understand P as something such as "A restaurant is known for their good food" and this would then imply their food (Q) will be good. Their food can still be good without these reviews as well (F -> T).

P	Q	P -> Q
T	T	T
T	F	F
F	T	T
F	F	T

As we can see through the truth table above P -> Q only resolves to false when P is True and Q is False. The implication is not effected by the value of Q when the value of P is False.

- Abduction is an inference rule that infers P from P → Q and Q. Show that abduction is not sound (give an example where it fails).

P	Q	P -> Q
T	T	T
T	F	F
F	T	T
F	F	T

As we can see through the truth table above abduction is not sound. In other words, we cannot prove that P -> Q means Q -> P because their conditions very. P -> Q holds true when pF qT, however T -> F is an invalid statement. Therefore we can conclude that P -> Q ≠ Q -> P.

- Show that Modus Tollens, from P → Q and ¬Q infer ¬P, is sound (here also use truth tables).

P	Q	P -> Q	¬Q infer ¬P	P → Q and ¬Q infer ¬P
T	T	T	T	T
T	F	F	F	T
F	T	T	T	T
F	F	T	T	T

As we can observe through the truth table above Modus Tollens does hold true. We can understand this by understanding the expanded form of P -> Q. I will use the example I previously provided.

$$(\neg P \land Q) \lor (\neg Q \land P)$$

We can now also provide the negated expanded form of the Modus Tollens alternative.

$$(\neg(\neg Q) \land (\neg P) \lor (\neg(\neg P) \land \neg Q)$$

This statement will then simplify down to the following:

$$(Q \land \neg P) \lor (P \land \neg Q)$$

Then because of the communicative law we can adjust our order of operations, thus proving their equivalence.

$$(\neg P \land Q) \lor (\neg Q \land P)$$