CSE 311 Section 2

Logic and Equivalences

Administrivia & Introductions

Announcements & Reminders

- Sections are Graded
  - You will be graded on section participation, so please try to come ☺
  - If you cannot attend you will need to submit ALL the section problems to gradescope

- HW1 due YESTERDAY **(4/9) @ 11:00 PM** on Gradescope
  - Homework is usually due **Wednesdays @ 11:00pm**, released Thursday evening
  - Remember, you only have 3 late days to use throughout the quarter
  - You can use only 1 late days on any 1 assignment

- Check the course website for OH times!

- Concept Checks!
  - Absolute deadline on **Thursdays @ 11:59 pm**

Task 1: Gates

Task 1

In this problem, we will represent NAND using function syntax, meaning NAND$(p, q) \equiv \neg(p \wedge q)$.

p	q	NAND(p, q)
T	T	F
T	F	T
F	T	T
F	F	T

We can construct any gate using only NAND gates [1]. For example, we can construct NOT by using the same input for both sides of a NAND.

a) Show $\neg p \equiv$NAND(p, p) holds using a truth table.

Task 1

a) Show $\neg p \equiv \mathrm{NAND}(p, p)$ holds using a truth table.

p	$\neg p$
T	F
F	T

Task 1

a) Show $\neg p \equiv \text{NAND}(p, p)$ holds using a truth table.

p	$\neg p$
T	F
F	T

Task 1

a) Show $\neg p \equiv \text{NAND}(p, p)$ holds using a truth table.

p
T
F

$\neg p$
F
T

p	$p \wedge p$
T	T
F	F

Task 1

a) Show $\neg p \equiv \mathrm{NAND}(p, p)$ holds using a truth table.

p	$\neg p$
T	F
F	T

p	$p \wedge p$	$\neg(p \wedge p)$
T	T	F
F	F	T

Task 1

a) Show $\neg p \equiv \text{NAND}(p, p)$ holds using a truth table.

p	$\neg p$
T	F
F	T

p	$p \wedge p$	$\neg(p \wedge p)$
T	T	F
F	F	T

Task 1

b) Show we can express $p \wedge q$ with only NAND gates by a chain of equivalences.

(p ∧ q) ≡ only NAND gates?

$$\mathsf{NAND}(p, q) \equiv \neg(p \wedge q).$$

$$\mathsf{NAND}(p, q) \equiv \neg(p \land q).$$

Task 1

b) Show we can express $p \land q$ with only NAND gates by a chain of equivalences.

$(p \land q) \equiv \neg\,\boxed{(\neg(p \land q) \land \neg(p \land q))}$

$$\text{NAND}(p, q) \equiv \neg(p \wedge q).$$

Task 1

b) Show we can express $p \wedge q$ with only NAND gates by a chain of equivalences.

(p ∧ q) ≡ ¬ (¬(p ∧ q) ∧ ¬(p ∧ q))

So our goal is: **NAND((NAND(p,q), NAND(p,q))**

Task 1

b) Show we can express $p \land q$ with only NAND gates by a chain of equivalences.

$$p \land q \equiv \neg\neg(p \land q) \qquad \text{Double Negation}$$

$$\equiv \mathrm{NAND}(\mathrm{NAND}(p, q), \mathrm{NAND}(p, q)) \qquad \text{Definition of NAND}$$

Task 1

b) Show we can express $p \wedge q$ with only NAND gates by a chain of equivalences.

$$\begin{aligned} p \wedge q &\equiv \neg\neg(p \wedge q) && \text{Double Negation} \\ &\equiv \neg\text{NAND}(p, q) && \text{Definition of NAND} \\ &\equiv \text{NAND}(\text{NAND}(p, q), \text{NAND}(p, q)) && \text{Definition of NAND} \end{aligned}$$

Task 1

b) Show we can express $p \wedge q$ with only NAND gates by a chain of equivalences.

$$
\begin{aligned}
p \wedge q &\equiv \neg\neg(p \wedge q) && \text{Double Negation} \\
&\equiv \neg\text{NAND}(p, q) && \text{Definition of NAND} \\
&\equiv \neg\text{NAND}(p, q) \vee \neg\text{NAND}(p, q) && \text{Idempotency} \\
&\equiv \text{NAND}(\text{NAND}(p, q), \text{NAND}(p, q)) && \text{Definition of NAND}
\end{aligned}
$$

Task 1

b) Show we can express $p \wedge q$ with only NAND gates by a chain of equivalences.

$$
\begin{aligned}
p \wedge q &\equiv \neg\neg(p \wedge q) && \text{Double Negation}\\
&\equiv \neg\mathsf{NAND}(p, q) && \text{Definition of NAND}\\
&\equiv \neg\mathsf{NAND}(p, q) \vee \neg\mathsf{NAND}(p, q) && \text{Idempotency}\\
&\equiv \neg(\mathsf{NAND}(p, q) \wedge \mathsf{NAND}(p, q)) && \text{De Morgan}\\
&\equiv \mathsf{NAND}(\mathsf{NAND}(p, q), \mathsf{NAND}(p, q)) && \text{Definition of NAND}
\end{aligned}
$$

$$\neg p \equiv \mathsf{NAND}(p, p)$$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

$(p \vee q) \equiv$ only NAND gates?

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: $\neg(p \vee q) \equiv \neg p \wedge \neg q$

$\neg p \equiv \mathrm{NAND}(p, p)$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: ¬(p ∨ q) ≡ ¬p ∧ ¬q

$\neg p \equiv \text{NAND}(p, p)$

(p ∨ q) ≡ *intuitively, how would you use these rules?*

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: $\neg(p \vee q) \equiv \neg p \wedge \neg q$

$\neg p \equiv \text{NAND}(p, p)$

$(p \vee q) \equiv \neg(\neg p \wedge \neg q)$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: ¬(p ∨ q) ≡ ¬p ∧ ¬q

$\neg p \equiv \text{NAND}(p, p)$

(p ∨ q) ≡ ¬(¬p ∧ ¬q)
≡ NAND(¬p, ¬q)

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: $\neg(p \vee q) \equiv \neg p \wedge \neg q$

$\neg p \equiv \text{NAND}(p, p)$

$$(p \vee q) \equiv \neg(\neg p \wedge \neg q)$$
$$\equiv \text{NAND}(\neg p, \neg q)$$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: $\neg(p \vee q) \equiv \neg p \wedge \neg q$

$\neg p \equiv \text{NAND}(p, p)$

$(p \vee q) \equiv \neg(\neg p \wedge \neg q)$ ← What is this step called?

$\equiv \text{NAND}(\neg p, \neg q)$

$\equiv \text{NAND}(\text{NAND}(p,p) \wedge \text{NAND}(q,q))$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

Remember:

Demorgan's: ¬(p ∨ q) ≡ ¬p ∧ ¬q

$\neg p \equiv \text{NAND}(p, p)$

(p ∨ q) ≡ ¬(¬p ∧ ¬q)

≡ NAND(¬p, ¬q)

≡ NAND(NAND(p,p) ∧ NAND(q,q))

This is actually **double negation** and then **Demorgan's!**

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

$$p \vee q \equiv \neg\neg(p \vee q) \qquad \text{Double Negation}$$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

$$
\begin{aligned}
p \vee q &\equiv \neg\neg(p \vee q) && \text{Double Negation} \\
&\equiv \neg(\neg p \wedge \neg q) && \text{De Morgan}
\end{aligned}
$$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \lor q$. Explain in English why your expression works.

$$
\begin{aligned}
p \lor q &\equiv \neg\neg(p \lor q) && \text{Double Negation} \\
&\equiv \neg(\neg p \land \neg q) && \text{De Morgan} \\
&\equiv \mathsf{NAND}(\neg p, \neg q) && \text{Definition of NAND}
\end{aligned}
$$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

$$
\begin{aligned}
p \vee q &\equiv \neg\neg(p \vee q) && \text{Double Negation} \\
&\equiv \neg(\neg p \wedge \neg q) && \text{De Morgan} \\
&\equiv \mathsf{NAND}(\neg p, \neg q) && \text{Definition of NAND} \\
&\equiv \mathsf{NAND}(\mathsf{NAND}(p, p), \neg q) && \text{Part a}
\end{aligned}
$$

Task 1

c) Use what we learned in (a) plus Double Negation to write an expression for $p \vee q$. Explain in English why your expression works.

$p \vee q \equiv \neg\neg(p \vee q)$	Double Negation
$\equiv \neg(\neg p \wedge \neg q)$	De Morgan
$\equiv \text{NAND}(\neg p, \neg q)$	Definition of NAND
$\equiv \text{NAND}(\text{NAND}(p, p), \neg q)$	Part a
$\equiv \text{NAND}(\text{NAND}(p, p), \text{NAND}(q, q))$	Part a

Task 3: Symbolic Proofs

Task 3 – Equivalences

b) $\neg p \to (q \to r) \equiv q \to (p \vee r)$

is this true?

Task 3 – Equivalences

b) $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

Yes! Truth values match!

p	q	r	$\neg p$	$(q \rightarrow r)$	$(p \vee r)$	$\neg p \rightarrow (q \rightarrow r)$	$q \rightarrow (p \vee r)$
T	T	T	F	T	T	T	T
T	T	F	F	F	T	T	T
T	F	T	F	T	T	T	T
T	F	F	F	T	T	T	T
F	T	T	T	T	T	T	T
F	T	F	T	F	F	F	F
F	F	T	T	T	T	T	T
F	F	F	T	T	F	T	T

Task 3 – Equivalences

b) $\neg p \to (q \to r) \equiv q \to (p \lor r)$

Remember these identities!

These identities hold for all propositions p, q, r

- Identity
  - $p \land \mathrm{T} \equiv p$
  - $p \lor \mathrm{F} \equiv p$
- Domination
  - $p \lor \mathrm{T} \equiv \mathrm{T}$
  - $p \land \mathrm{F} \equiv \mathrm{F}$
- Idempotent
  - $p \lor p \equiv p$
  - $p \land p \equiv p$
- Commutative
  - $p \lor q \equiv q \lor p$
  - $p \land q \equiv q \land p$
- Associative
  - $(p \lor q) \lor r \equiv p \lor (q \lor r)$
  - $(p \land q) \land r \equiv p \land (q \land r)$
- Distributive
  - $p \land (q \lor r) \equiv (p \land q) \lor (p \land r)$
  - $p \lor (q \land r) \equiv (p \lor q) \land (p \lor r)$
- Absorption
  - $p \lor (p \land q) \equiv p$
  - $p \land (p \lor q) \equiv p$
- Negation
  - $p \lor \neg p \equiv \mathrm{T}$
  - $p \land \neg p \equiv \mathrm{F}$
- DeMorgan's Laws
  - $\neg(p \lor q) \equiv \neg p \land \neg q$
  - $\neg(p \land q) \equiv \neg p \lor \neg q$
- Double Negation
  - $\neg\neg p \equiv p$
- Law of Implication
  - $p \to q \equiv \neg p \lor q$
- Contrapositive
  - $p \to q \equiv \neg q \to \neg p$

Task 3 – Equivalences

b) $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

Task 3 – Equivalences

b) $\neg p \to (q \to r) \equiv q \to (p \vee r)$

$$
\begin{aligned}
\neg p \to (q \to r) \quad & \equiv \quad \neg\neg p \vee (q \to r) && \text{Law of Implication} \\
& \equiv \quad \neg q \vee (p \vee r) && \\
& \equiv \quad q \to (p \vee r) && \text{Law of Implication}
\end{aligned}
$$

Task 3 – Equivalences

b) $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

$$
\begin{aligned}
\neg p \rightarrow (q \rightarrow r) \quad & \equiv \quad \neg\neg p \vee (q \rightarrow r) && \text{Law of Implication} \\
& \equiv \quad p \vee (q \rightarrow r) && \text{Double Negation} \\
& \equiv \quad \neg q \vee (p \vee r) && \\
& \equiv \quad q \rightarrow (p \vee r) && \text{Law of Implication}
\end{aligned}
$$

Task 3 – Equivalences

b) $\neg p \to (q \to r) \equiv q \to (p \vee r)$

$$
\begin{array}{lcll}
\neg p \to (q \to r) & \equiv & \neg\neg p \vee (q \to r) & \text{Law of Implication} \\
 & \equiv & p \vee (q \to r) & \text{Double Negation} \\
 & \equiv & p \vee (\neg q \vee r) & \text{Law of Implication} \\
 & & & \\
 & \equiv & \neg q \vee (p \vee r) & \\
 & \equiv & q \to (p \vee r) & \text{Law of Implication}
\end{array}
$$

Task 3 – Equivalences

b) $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

$\neg p \rightarrow (q \rightarrow r)$	$\equiv$	$\neg\neg p \vee (q \rightarrow r)$	Law of Implication
	$\equiv$	$p \vee (q \rightarrow r)$	Double Negation
	$\equiv$	$p \vee (\neg q \vee r)$	Law of Implication
	$\equiv$	$(p \vee \neg q) \vee r$	Associativity
	$\equiv$	$\neg q \vee (p \vee r)$	
	$\equiv$	$q \rightarrow (p \vee r)$	Law of Implication

Task 3 – Equivalences

b) $\neg p \to (q \to r) \equiv q \to (p \vee r)$

$\neg p \to (q \to r)$	$\equiv$	$\neg\neg p \vee (q \to r)$	Law of Implication
	$\equiv$	$p \vee (q \to r)$	Double Negation
	$\equiv$	$p \vee (\neg q \vee r)$	Law of Implication
	$\equiv$	$(p \vee \neg q) \vee r$	Associativity
	$\equiv$	$(\neg q \vee p) \vee r$	Commutativity
	$\equiv$	$\neg q \vee (p \vee r)$	
	$\equiv$	$q \to (p \vee r)$	Law of Implication

Task 3 – Equivalences

b) $\neg p \rightarrow (q \rightarrow r) \equiv q \rightarrow (p \vee r)$

$$
\begin{array}{llll}
\neg p \rightarrow (q \rightarrow r) & \equiv & \neg\neg p \vee (q \rightarrow r) & \text{Law of Implication} \\
 & \equiv & p \vee (q \rightarrow r) & \text{Double Negation} \\
 & \equiv & p \vee (\neg q \vee r) & \text{Law of Implication} \\
 & \equiv & (p \vee \neg q) \vee r & \text{Associativity} \\
 & \equiv & (\neg q \vee p) \vee r & \text{Commutativity} \\
 & \equiv & \neg q \vee (p \vee r) & \text{Associativity} \\
 & \equiv & q \rightarrow (p \vee r) & \text{Law of Implication}
\end{array}
$$

Task 5: Translate to English

Translate these system specifications into English where $F(p)$ is "Printer p is out of service", $B(p)$ is "Printer p is busy", $L(j)$ is "Print job j is lost," and $Q(j)$ is "Print job j is queued". Let the domain be all printers and all print jobs.

a) $\exists p\,(F(p) \wedge B(p)) \rightarrow \exists j\; L(j)$

b) $(\forall j\; B(j)) \rightarrow (\exists p\; Q(p))$

Translate these system specifications into English where $F(p)$ is “Printer p is out of service”, $B(p)$ is “Printer p is busy”, $L(j)$ is “Print job j is lost,” and $Q(j)$ is “Print job j is queued”. Let the domain be all printers and all print jobs.

a) $\exists p\,(F(p) \wedge B(p)) \rightarrow \exists j\, L(j)$

If at least one printer is busy and out of service, then at least one job is lost.

b) $(\forall j\, B(j)) \rightarrow (\exists p\, Q(p))$

Translate these system specifications into English where $F(p)$ is "Printer p is out of service", $B(p)$ is "Printer p is busy", $L(j)$ is "Print job j is lost," and $Q(j)$ is "Print job j is queued". Let the domain be all printers and all print jobs.

a) $\exists p\,(F(p) \wedge B(p)) \rightarrow \exists j\, L(j)$

If at least one printer is busy and out of service, then at least one job is lost.

b) $(\forall j\, B(j)) \rightarrow (\exists p\, Q(p))$

If all printers are busy, then there is a queued job.

Task 6: Translate to Logic

Task 6

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

a) Every user has access to an electronic mailbox.

b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Task 6

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

a) Every user has access to an electronic mailbox.

Let the domain be users and mailboxes. Let $\texttt{User}(x)$ be "x is a user", let $\texttt{Mailbox}(y)$ be "y is a mailbox", and let $\texttt{Access}(x, y)$ be "x has access to y".

Task 6

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

a) Every user has access to an electronic mailbox.

Let the domain be users and mailboxes. Let $\texttt{User}(x)$ be "x is a user", let $\texttt{Mailbox}(y)$ be "y is a mailbox", and let $\texttt{Access}(x, y)$ be "x has access to y".

$$\forall x\ (\texttt{User}(x) \rightarrow (\exists y\ (\texttt{Mailbox}(y) \wedge \texttt{Access}(x, y))))$$

Task 6

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Task 6

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Let the domain be people and mailboxes and use $\texttt{Access}(x, y)$ as defined in the solution to part (a), and then also add $\texttt{InGroup}(x)$ for "x is in the group", and let $\texttt{SystemMailBox}$ be the name for the system mailbox. Then the translation becomes

$$\texttt{FileSystemLocked} \to \forall x\ (\texttt{InGroup}(x) \to \texttt{Access}(x, \texttt{SystemMailBox})).$$

Task 6

Express each of these system specifications using predicates, quantifiers, and logical connectives. For some of these problems, more than one translation will be reasonable depending on your choice of predicates.

b) The system mailbox can be accessed by everyone in the group if the file system is locked.

Let the domain be people in the group. Let $\texttt{CanAccessSM}(x)$ be "x has access to the system mailbox". Let p be the proposition "the file system is locked."

$$p \to \forall x \ \texttt{CanAccessSM}(x).$$

That’s All, Folks!

Thanks for coming to section this week!
Any questions?

Task 4: CNF and Simplification

Task 4b – Canonical Forms

b) Write the **CNF** expressions for G(A,B,C)

A	*B*	*C*	*G(A,B,C)*
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

Work on part (b) with the people around you, and then we'll go over it together!

Task 4b – Canonical Forms

b) Write the **CNF** expressions for G(A,B,C)

A	*B*	*C*	*G(A,B,C)*
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	0
0	1	1	1
0	1	0	0
0	0	1	1
0	0	0	0

(A'+B+C)

(A+B'+C)

(A+B+C)

(A'+B+C)(A+B'+C)(A+B+C)

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

Identity
$A \wedge \mathsf{T} \equiv A$
$A \vee \mathsf{F} \equiv A$

Domination
$A \vee \mathsf{T} \equiv \mathsf{T}$
$A \wedge \mathsf{F} \equiv \mathsf{F}$

Idempotency
$A \vee A \equiv A$
$A \wedge A \equiv A$

Commutativity
$A \vee B \equiv B \vee A$
$A \wedge B \equiv B \wedge A$

Associativity
$(A \vee B) \vee C \equiv A \vee (B \vee C)$
$(A \wedge B) \wedge C \equiv A \wedge (B \wedge C)$

Distributivity
$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$
$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

Absorption
$A \vee (A \wedge B) \equiv A$
$A \wedge (A \vee B) \equiv A$

Negation
$A \vee \neg A \equiv \mathsf{T}$
$A \wedge \neg A \equiv \mathsf{F}$

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$$ Idempotency

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$$ Idempotency

$$= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C)$$ Commutativity x 2

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$$
\begin{aligned}
&= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C) && \text{Idempotency} \\
&= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C) && \text{Commutativity x 2} \\
&= (A' + B + C)(A + B + C)(A + C + B')(A + C + B) && \text{Commutativity x 2}
\end{aligned}
$$

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$$ Idempotency

$$= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C)$$ Commutativity x 2

$$= (A' + B + C)(A + B + C)(A + C + B')(A + C + B)$$ Commutativity x 2

$$= (B + C + A')(B + C + A)(A + C + B')(A + C + B)$$ Commutativity x 2

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$	Idempotency
$= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C)$	Commutativity x 2
$= (A' + B + C)(A + B + C)(A + C + B')(A + C + B)$	Commutativity x 2
$= (B + C + A')(B + C + A)(A + C + B')(A + C + B)$	Commutativity x 2
$= [(B + C) + (A' \cdot A)][(A + C) + (B' \cdot B)]$	Distributivity x2

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$	Idempotency
$= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C)$	Commutativity x 2
$= (A' + B + C)(A + B + C)(A + C + B')(A + C + B)$	Commutativity x 2
$= (B + C + A')(B + C + A)(A + C + B')(A + C + B)$	Commutativity x 2
$= [(B + C) + (A' \cdot A)][(A + C) + (B' \cdot B)]$	Distributivity x2
$= [(B + C) + 0][(A + C) + 0]$	Negation x 2

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$$
\begin{aligned}
&= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C) && \text{Idempotency} \\
&= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C) && \text{Commutativity x 2} \\
&= (A' + B + C)(A + B + C)(A + C + B')(A + C + B) && \text{Commutativity x 2} \\
&= (B + C + A')(B + C + A)(A + C + B')(A + C + B) && \text{Commutativity x 2} \\
&= [(B + C) + (A' \cdot A)][(A + C) + (B' \cdot B)] && \text{Distributivity x2} \\
&= [(B + C) + 0][(A + C) + 0] && \text{Negation x 2} \\
&= [B + C][A + C] && \text{Identity}
\end{aligned}
$$

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$	Idempotency
$= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C)$	Commutativity x 2
$= (A' + B + C)(A + B + C)(A + C + B')(A + C + B)$	Commutativity x 2
$= (B + C + A')(B + C + A)(A + C + B')(A + C + B)$	Commutativity x 2
$= [(B + C) + (A' \cdot A)][(A + C) + (B' \cdot B)]$	Distributivity x2
$= [(B + C) + 0][(A + C) + 0]$	Negation x 2
$= [B + C][A + C]$	Identity
$= [C + B][C + A]$	Commutativity

Task 4c – Canonical Forms

$$(A' + B + C)(A + B' + C)(A + B + C)$$

$= (A' + B + C)(A + B' + C)(A + B + C)(A + B + C)$	Idempotency
$= (A' + B + C)(A + B + C)(A + B' + C)(A + B + C)$	Commutativity x 2
$= (A' + B + C)(A + B + C)(A + C + B')(A + C + B)$	Commutativity x 2
$= (B + C + A')(B + C + A)(A + C + B')(A + C + B)$	Commutativity x 2
$= [(B + C) + (A' \cdot A)][(A + C) + (B' \cdot B)]$	Distributivity x2
$= [(B + C) + 0][(A + C) + 0]$	Negation x 2
$= [B + C][A + C]$	Identity
$= [C + B][C + A]$	Commutativity
$= C + B \cdot A$	Distributivity