Study Guide for MAT 243 Review Exam 1

# Symbols and Logical Operators

• p, q, r, a: Propositional variables representing statements.

• → : Implies (if ... then).

• ∧ : Logical AND (conjunction).

• ∨ : Logical OR (disjunction).

• ¬ : Logical NOT (negation).

• ∀ : Universal quantifier (for all).

• ∃ : Existential quantifier (there exists).

• ↔ : Biconditional (if and only if).

# 1. Proposition: Is 'Frederic Gauss is the greatest mathematician of all time' a Proposition?

Key Idea: In many logic texts, a proposition is defined as a statement that is either true or false. A proper proposition must have an objectively verifiable truth value.

According to your professor, the statement is an opinion that cannot be objectively verified, and therefore, it is not considered a proposition.

# 2. Important Logical Equivalences

## a. Conditional as a Disjunction

Expression: p → q ≡ ¬p ∨ q

Plain Language: 'If p then q' means either p is false or q is true.

## b. Contrapositive of a Conditional Statement

Expression: p → q ≡ ¬q → ¬p

Plain Language: If 'p implies q' is true, then 'if q is false, then p must be false' is also true.

## c. Negation of a Conditional Statement

Expression: ¬(p → q) ≡ p ∧ ¬q

Plain Language: Saying 'it is not true that if p then q' is equivalent to 'p is true and q is false'.

## d. De Morgan’s Identities

For Conjunction: ¬(p ∧ q) ≡ ¬p ∨ ¬q

For Disjunction: ¬(p ∨ q) ≡ ¬p ∧ ¬q

Plain Language: These identities explain how negation distributes over 'and' and 'or'.

# 3. Prove that (p ∧ q) → r ≡ (p → r) ∨ (q → r)

Goal: Prove that (p ∧ q) → r is logically equivalent to (p → r) ∨ (q → r).

Step-by-Step:

1. Start with (p ∧ q) → r.

2. Rewrite as a disjunction: (p ∧ q) → r ≡ ¬(p ∧ q) ∨ r.

3. Apply De Morgan’s Law: ¬(p ∧ q) ≡ ¬p ∨ ¬q, so the expression becomes ¬p ∨ ¬q ∨ r.

4. Note that p → r ≡ ¬p ∨ r and q → r ≡ ¬q ∨ r.

5. Therefore, (p → r) ∨ (q → r) ≡ (¬p ∨ r) ∨ (¬q ∨ r), which simplifies (by associativity/commutativity) to ¬p ∨ ¬q ∨ r.

6. Conclusion: Since both sides simplify to the same expression, the equivalence holds.

# 4. Express the Negation of the Following Statements

## a. Negate: ∀x ∃y (xy ≤ 0 → (-1 ≤ y < 0))

Steps:

• Negate the entire statement: ¬[∀x ∃y (xy ≤ 0 → (-1 ≤ y < 0))]

• Change the quantifiers: ≡ ∃x ∀y ¬(xy ≤ 0 → (-1 ≤ y < 0))

• Negate the conditional using ¬(A → B) ≡ A ∧ ¬B: ¬(xy ≤ 0 → (-1 ≤ y < 0)) ≡ (xy ≤ 0) ∧ ¬(-1 ≤ y < 0)

• Final Form: ∃x ∀y [(xy ≤ 0) ∧ ¬(-1 ≤ y < 0)]

## b. Negate: 'Every planet in this solar system revolves around the Sun.'

Steps:

• Express in logical form: ∀x (P(x) → R(x)), where

P(x): 'x is a planet in this solar system'

R(x): 'x revolves around the Sun'

• Negate and change quantifiers: ¬∀x (P(x) → R(x)) ≡ ∃x ¬(P(x) → R(x))

• Negate the conditional: ¬(P(x) → R(x)) ≡ P(x) ∧ ¬R(x)

• Final Form: ∃x [P(x) ∧ ¬R(x)]

## c. Negate: 'Some people in Arizona don’t like to drive on highways.'

Steps:

• Logical form: ∃x [A(x) ∧ ¬D(x)], where

A(x): 'x is a person in Arizona'

D(x): 'x likes to drive on highways'

• Negate the statement: ¬∃x [A(x) ∧ ¬D(x)] ≡ ∀x ¬[A(x) ∧ ¬D(x)]

• Apply De Morgan’s Law: ¬[A(x) ∧ ¬D(x)] ≡ ¬A(x) ∨ D(x)

• Final Form: ∀x [¬A(x) ∨ D(x)]

# 5. Translating with 'Necessary', 'Sufficient', 'Only If', and 'Unless'

Original Statement: 'If there are no clouds, then I can see the stars.'

Let C = 'there are clouds' and S = 'I can see the stars'.

Direct form: ¬C → S

Alternate phrasings:

• Only if: 'I can see the stars only if there are no clouds.' (Implies: S → ¬C)

• Necessary: 'For me to see the stars, it is necessary that there are no clouds.'

• Sufficient: 'The absence of clouds is sufficient for me to see the stars.'

• Unless: 'I can see the stars unless there are clouds.' (Interpreted as: if there are clouds then I cannot see the stars, i.e., C → ¬S)

# 6. Expressing Statements in 'If... then' Form

## a. 'I get my work done on time only if I don’t procrastinate.'

Let W = 'work is done on time' and P = 'I procrastinate'.

Translation: W → ¬P

## b. 'It is necessary to have wind to sail across the lake.'

Let S = 'sail across the lake' and W = 'have wind'.

Translation: S → W

## c. 'To be able to vote in the upcoming election it is sufficient to have an American passport.'

Let V = 'able to vote' and P = 'have an American passport'.

Translation: P → V

## d. 'Unless we live more economically we are going to use up all our resources.'

Let E = 'live more economically' and U = 'use up all resources'.

Translation: ¬E → U

# 7. Expressing Conditionals as Disjunctions

## a. Rewrite ¬p → ¬q

Using the equivalence: p → q ≡ ¬p ∨ q

Replace p with ¬p and q with ¬q:

¬p → ¬q ≡ ¬(¬p) ∨ ¬q ≡ p ∨ ¬q

## b. Rewrite a → ¬p

Using the equivalence:

a → ¬p ≡ ¬a ∨ ¬p

# 8. Expressing Disjunction as a Conditional

Express p ∨ q using a conditional:

p ∨ q ≡ ¬p → q

Plain Language: 'If not p then q' is equivalent to 'p or q.'

# 9. Expressing Conjunction Using Conditional and Negation

Express p ∧ ¬q:

Observation: The negation of a conditional yields a conjunction, i.e., ¬(p → q) ≡ p ∧ ¬q

Thus, p ∧ ¬q ≡ ¬(p → q)

# 10. Expressing ¬p ∧ ¬q Using Conditional and Negation

Steps:

1. Start with De Morgan’s Law: ¬p ∧ ¬q ≡ ¬(p ∨ q)

2. Express the disjunction as a conditional: p ∨ q ≡ ¬p → q

3. Then, ¬(p ∨ q) ≡ ¬(¬p → q)

Conclusion: ¬p ∧ ¬q ≡ ¬(¬p → q)

# 11. Evaluating Arguments

## Argument 1: Valid

Premises:

1. Every bug is an insect: ∀x (B(x) → I(x))

2. Elephants are not insects: ∀x (E(x) → ¬I(x))

3. Trombone is an elephant: E(Trombone)

Conclusion: ¬B(Trombone)

Explanation: Since every bug must be an insect and Trombone (an elephant) is not an insect, it cannot be a bug.

## Argument 2: Invalid

Premises:

1. Every bug is an insect: ∀x (B(x) → I(x))

2. Wizzy-Fizz is an insect: I(Wizzy-Fizz)

Conclusion: B(Wizzy-Fizz)

Explanation: Although every bug is an insect, not every insect is a bug. This mistake is known as affirming the consequent.

# Final Tips

• Definitions matter – always use the definitions your course provides.

• Practice converting statements using logical equivalences.

• Write each step in proofs clearly so you can follow the logic.

• Identify logical fallacies by checking if conclusions follow from the premises.