MTH202 · DISCRETE MATHEMATICS

# Lecture 2

Larger truth tables, logical equivalence, De Morgan's laws and tautology.

## What you will learn

| | |
|---|---|
| Step 1 | Building larger truth tables (2 and 3 variables) |
| Step 2 | Logical equivalence (≡) and the double negative property |
| Step 3 | De Morgan's laws, for sentences and for inequalities |
| Step 4 | Tautology |

**Before you begin**

This lecture builds on the truth tables of Lecture 1. Quick reminder: **~** flips the value, ∧ is T only for TT, ∨ is F only for FF.

# Step 1: Building larger truth tables

**Key idea**

Never try to solve a long statement in one go. **Break it into small parts** and give **each part its own column**. The last column is the answer.

## How many rows?

| Number of letters | Rows | Working |
|---|---|---|
| 1 (p) | 2 | $2^1 = 2$ |
| 2 (p, q) | 4 | $2^2 = 4$ |
| 3 (p, q, r) | 8 | $2^3 = 8$ |
| 4 (p, q, r, s) | 16 | $2^4 = 16$ |

**Writing the 8 rows**

Rows = $\mathbf{2^n}$, where n is the number of different letters.

For 8 rows, fill the columns like this:

- **p**: four T, then four F
- **q**: two T, two F, two T, two F
- **r**: T, F, T, F, ... (one by one)

### Example 1: ~p ∧ q

Parts: first find **~p**, then **~p ∧ q**.

| p | q | ~p | ~p ∧ q |
|---|---|---|---|
| T | T | F | **F** |
| T | F | F | **F** |
| F | T | T | **T** |
| F | F | T | **F** |

Only row 3 is true, because ∧ needs **both** ~p and q to be T.

### Example 2: ~p ∧ (q ∨ ~r)

Parts: **~r**, then **q ∨ ~r**, then **~p**, and finally **~p ∧ (q ∨ ~r)**.

| p | q | r | ~r | q ∨ ~r | ~p | ~p ∧ (q ∨ ~r) |
|---|---|---|---|---|---|---|
| T | T | T | F | T | F | **F** |
| T | T | F | T | T | F | **F** |
| T | F | T | F | F | F | **F** |
| T | F | F | T | T | F | **F** |
| F | T | T | F | T | T | **T** |
| F | T | F | T | T | T | **T** |
| F | F | T | F | F | T | **F** |
| F | F | F | T | T | T | **T** |

Notice: whenever p is T, ~p is F, so the whole statement is F. That is why the first four rows are all F.

### Example 3: (p ∨ q) ∧ ~(p ∧ q)

Parts: **p ∨ q**, **p ∧ q**, **~(p ∧ q)**, then join with ∧.

| p | q | p ∨ q | p ∧ q | ~(p ∧ q) | (p ∨ q) ∧ ~(p ∧ q) |
|---|---|---|---|---|---|
| T | T | T | T | F | **F** |
| T | F | T | F | T | **T** |
| F | T | T | F | T | **T** |
| F | F | F | F | T | **F** |

> **Key idea**
> This statement is true exactly when **only one** of p and q is true, not both.

**Practice**

1. How many rows does a truth table with 4 letters (p, q, r, s) have?
2. If p = F and q = T, what is ~p ∧ q?
3. If p = T and q = T, what is (p ∨ q) ∧ ~(p ∧ q)?
4. If p = F, q = F and r = T, what is ~p ∧ (q ∨ ~r)?

**Answers (cover them first, then check)**

1. $2^4$ = 16 rows.
2. T (~p = T, and T ∧ T = T).
3. F (~(p ∧ q) = F, so the ∧ is F).
4. F (~r = F, so q ∨ ~r = F ∨ F = F; ~p = T; T ∧ F = F).

# Step 2: Logical equivalence and the double negative

## Start with an everyday example

"Ali **and** Ahmed went to school" and "Ahmed **and** Ali went to school". The word order is different, but the **meaning is the same**. Whenever one is true, the other is true; whenever one is false, the other is false.

**Key idea**

Two statement forms are **logically equivalent** when they have the **same truth value in every row** of their truth tables.

Symbol: ≡, read "is logically equivalent to".

Let p = "Ali went to school" and q = "Ahmed went to school". Compare p ∧ q with q ∧ p:

| p | q | p ∧ q | q ∧ p | Same? |
|---|---|---|---|---|
| T | T | **T** | **T** | Yes |
| T | F | **F** | **F** | Yes |
| F | T | **F** | **F** | Yes |
| F | F | **F** | **F** | Yes |

All four rows match, so **p ∧ q ≡ q ∧ p**.

Now compare p ∧ q ("Ali **and** Ahmed went") with p ∨ q ("Ali **or** Ahmed went"):

| p | q | p ∧ q | p ∨ q | Same? |
|---|---|---|---|---|
| T | T | **T** | **T** | Yes |
| T | F | **F** | **T** | No |
| F | T | **F** | **T** | No |
| F | F | **F** | **F** | Yes |

Rows 2 and 3 are different, so p ∧ q and p ∨ q are **not** equivalent.

**How to check**

Make the truth table, look at the two final columns, and compare row by row.

- Every row the same: **equivalent (≡)**.
- Even **one** row different: **not equivalent**.

## The double negative property: ~(~p) ≡ p

Negating twice brings you back to where you started.

| p | ~p | ~(~p) |
|---|---|---|
| **T** | F | **T** |
| **F** | T | **F** |

**Example**

"It is not true that I am not happy."

Let p = "I am happy". Then ~p = "I am not happy", and the sentence is ~(~p).

Since ~(~p) ≡ p, the sentence simply means: **"I am happy"**.

## A pair that is NOT equivalent: ~(p ∧ q) and ~p ∧ ~q

| p | q | ~p | ~q | p ∧ q | ~(p ∧ q) | ~p ∧ ~q |
|---|---|---|---|---|---|---|
| T | T | F | F | T | **F** | **F** |
| T | F | F | T | F | **T** | **F** |
| F | T | T | F | F | **T** | **F** |
| F | F | T | T | F | **T** | **T** |

Rows 2 and 3 are different, so they are **not** equivalent.

**Exam trap**

Do not just say "because of the brackets". The correct reason is: **their truth tables differ in rows 2 and 3**.

In plain words, with p = "Ali went" and q = "Ahmed went":

- ~(p ∧ q) = "It is not the case that **both** went" (at least one did not go).
- ~p ∧ ~q = "Ali did not go **and** Ahmed did not go" (nobody went).

If Ali went and Ahmed did not (row 2), the first is T but the second is F.

## Exercises from the handout

- **(p ∧ q) ∧ r ≡ p ∧ (q ∧ r)** is true: both sides are T only when p, q and r are all T, so all 8 rows match.
- **(p ∧ q) ∨ r** and **p ∧ (q ∨ r)** are **not** equivalent. Try p = F, q = F, r = T: the first is F ∨ T = T, but the second is F ∧ T = F. One different row is enough.

**Practice**

1. When are two statement forms logically equivalent?
2. Are p ∨ q and q ∨ p equivalent?
3. Two statement forms match in 3 rows and differ in 1 row. Are they equivalent?
4. What is ~(~q) equal to?
5. Simplify: "It is not true that Ali is not a student."

**Answers (cover them first, then check)**

1. When they have the same truth value in every row of the truth table.
2. Yes ("Ali or Ahmed" means the same as "Ahmed or Ali").
3. No. Even one different row means not equivalent.
4. q
5. Ali is a student.

# Step 3: De Morgan's laws

In Step 2 we saw that ~(p ∧ q) is **not** ~p ∧ ~q. So what is it equal to? De Morgan's laws answer this.

| Law | In words | In symbols |
|---|---|---|
| 1 | The negation of an **and** statement is the **or** statement with each part negated. | **~(p ∧ q) ≡ ~p ∨ ~q** |
| 2 | The negation of an **or** statement is the **and** statement with each part negated. | **~(p ∨ q) ≡ ~p ∧ ~q** |

**The two-step trick**

When ~ goes inside the brackets, do **two** things:

- Put **~ on every part**.
- **Swap the connective**: ∧ becomes ∨, and ∨ becomes ∧.

$$\sim( p \wedge q ) \rightarrow \sim p \vee \sim q$$

## Proof of law 2 by truth table: ~(p ∨ q) ≡ ~p ∧ ~q

| p | q | ~p | ~q | p ∨ q | ~(p ∨ q) | ~p ∧ ~q |
|---|---|---|---|---|---|---|
| T | T | F | F | T | **F** | **F** |
| T | F | F | T | T | **F** | **F** |
| F | T | T | F | T | **F** | **F** |
| F | F | T | T | F | **T** | **T** |

The two final columns match in every row, so they are logically equivalent.

## Application 1: negating English sentences

| Statement | Negation |
|---|---|
| The fan is slow **or** it is very hot. | The fan is **not** slow **and** it is **not** very hot. |
| Akram is unfit **and** Saleem is injured. | Akram is **not** unfit **or** Saleem is **not** injured. |
| Ali is tall **and** Ahmed is short. | Ali is **not** tall **or** Ahmed is **not** short. |
| It is cold **or** it is raining. | It is **not** cold **and** it is **not** raining. |

**Exam trap**

The most common mistake is adding "not" but **forgetting to swap and/or**.

Wrong: "Ali is not tall **and** Ahmed is not short." Right: "Ali is not tall **or** Ahmed is not short."

## Application 2: inequalities

Question: write the negation of **-1 < x ≤ 4** for a real number x.

**Key idea**

A double inequality has a **hidden "and"**.

-1 < x ≤ 4 means **x > -1 and x ≤ 4**: x lies between -1 and 4.




**Step a.** Break it up: x > -1 **and** x ≤ 4.

**Step b.** Apply De Morgan: swap **and** to **or**, and negate each part.

| Part | Its negation |
|---|---|
| x > -1 | x ≤ -1 |
| x ≤ 4 | x > 4 |

**Answer:** the negation is **x ≤ -1 or x > 4**.




Why **or**? A number cannot be at most -1 **and** greater than 4 at the same time. It is on one side **or** the other.

| x | -1 < x ≤ 4 ? | x ≤ -1 or x > 4 ? |
|---|---|---|
| 2 | T | F |
| -3 | F | T |
| 7 | F | T |

In every row the two answers are opposite, which confirms the negation is correct.

**Flipping inequality signs**

The opposite of **>** is ≤, and the opposite of ≤ is **>**.

The opposite of **<** is ≥, and the opposite of ≥ is **<**.

**Exam trap: "means" vs "negation"**

Read the question word carefully before choosing an option.

- "-1 < x ≤ 4 **means** ...": only break it up. Answer: **x > -1 and x ≤ 4**.
- "The **negation** of -1 < x ≤ 4 is ...": break it up and apply De Morgan. Answer: **x ≤ -1 or x > 4**.

Words like *negation, negate, not true* mean flip it. Words like *means, equivalent to, same as* mean do not flip anything.

**Note**

One middle line of the handout prints "x > -1 or x ≤ 4". That is a printing slip; the handout's final answer, **x ≤ -1 or x > 4**, is correct.

**Practice**

1. Complete: ~(p ∧ q) ≡ ?
2. Write the negation of "Sara is happy and Hina is sad".
3. Write the negation of "The shop is open or the road is closed".
4. What does 3 < x ≤ 9 mean?
5. Write the negation of 2 < x ≤ 7.

**Answers (cover them first, then check)**

1. ~p ∨ ~q
2. Sara is not happy or Hina is not sad.
3. The shop is not open and the road is not closed.
4. x > 3 and x ≤ 9.
5. x ≤ 2 or x > 7 (do not use a comma; write "or").

# Step 4: Tautology

**Key idea**

A **tautology** is a statement form that is **always true**, whatever the truth values of its variables.

In its truth table, the final column contains **only T**.

A tautology is represented by the symbol **t**.

### Example: p ∨ ~p

| p | ~p | p ∨ ~p |
|---|---|---|
| T | F | **T** |

| p | ~p | p ∨ ~p |
|---|---|---|
| F | T | **T** |

Both rows are T, so **p ∨ ~p ≡ t**.

**Example**

Let p = "It will rain today". Then p ∨ ~p = "It will rain today **or** it will not rain today". Whatever happens, one of the two must be true, so the sentence is always true.

**How to check**

- Make the truth table.
- Look at the final column.
- All T: **tautology**. Even one F: **not a tautology**.

**Practice**

1. What symbol represents a tautology?
2. Is "Ali is a student or Ali is not a student" a tautology? Why?
3. Is p ∧ q a tautology?
4. Is q ∨ ~q a tautology?

**Answers (cover them first, then check)**

1. t
2. Yes. One of the two parts is always true (it has the form p ∨ ~p).
3. No. Its final column has F in three rows (only TT gives T).
4. Yes. Its final column is all T.

# Revision sheet: Lecture 2

| Topic | What to remember |
|---|---|
| Rows in a truth table | $2^n$ (2 letters: 4 rows, 3 letters: 8 rows) |
| Large statements | Break into parts; one column per part; last column is the answer. |
| Logical equivalence (≡) | Same truth value in every row. One different row: not equivalent. |
| Double negative | ~(~p) ≡ p |
| Common trap | ~(p ∧ q) is NOT ≡ ~p ∧ ~q (rows 2 and 3 differ). |
| De Morgan law 1 | ~(p ∧ q) ≡ ~p ∨ ~q |
| De Morgan law 2 | ~(p ∨ q) ≡ ~p ∧ ~q |
| De Morgan trick | Put ~ on each part and swap and ↔ or. |
| Inequalities | a < x ≤ b means x > a and x ≤ b; its negation is x ≤ a or x > b. |
| "means" vs "negation" | means: just break it up. negation: break it up and flip. |
| Tautology (t) | Always true; final column all T. Example: p ∨ ~p ≡ t. |