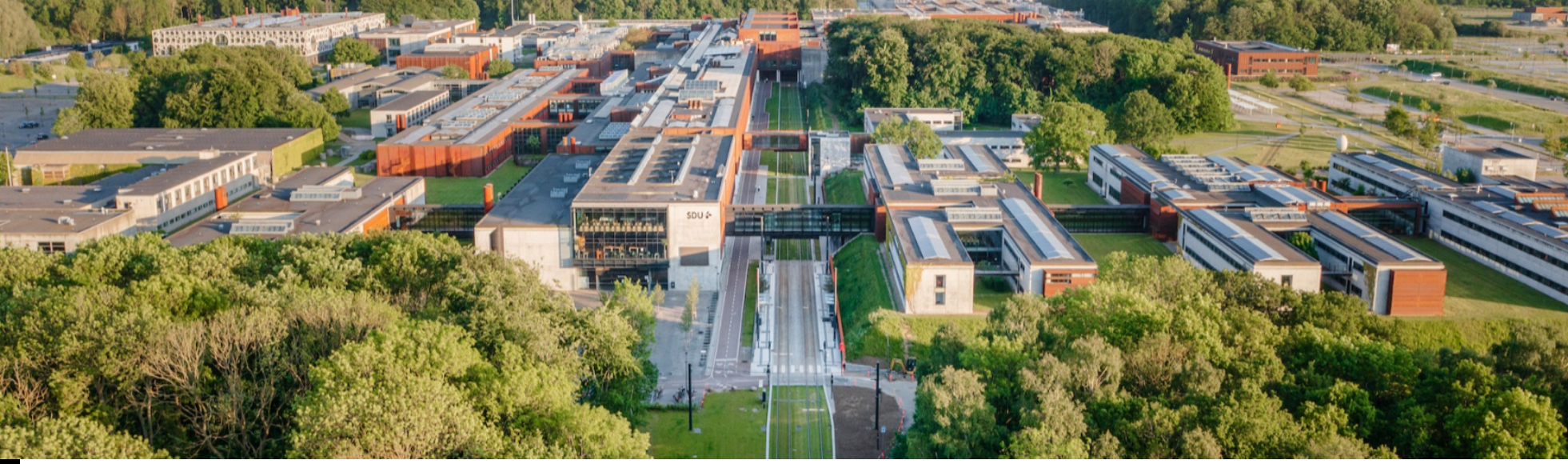

Brush-Up Maths for Data Science (2025)

Lecture Slides, Aug. 30th

Nicklas S. Andersen

University of Southern Denmark (SDU)
Department of Mathematics & Computer Science (IMADA)

Mathematical Logic

Mathematical logic is the branch of mathematics that:

- Studies reasoning using precise rules and symbols
- Represents and manipulates statements logically

These principles are not purely theoretical, they form the foundation for:

- Digital circuits in computers (hardware)
- Conditional logic in programming (software)

In the following, we will explore:

- Propositions and their truth values
- Logical operations like AND, OR, NOT, and IF…THEN…
- Truth tables, tautologies, and contradictions
- Logical equivalences and implications

Propositions

A proposition is a declarative statement that can be assigned a truth value: true or false, but never both.

In mathematical logic:

- Propositions reflect everyday reasoning but with strict precision.
- They can be simple or compound, formed using logical connectives like and, or, and not.

Understanding propositions forms the basis for:

- Conditional statements in programming (e.g., involving `` `if` `` and `` `else` ``)
- Formal mathematical proofs

Examples:

- "Four is even." (True)
- "1 + 1 is 3." (False)
- "$43 > 21$." (True)
- "$4 \in \{1, 3, 5\}$." (False)

Logical Operations

Simple propositions can be combined into compound propositions using logical connectives such as:

- AND
- OR
- NOT
- IF…THEN…
- IF AND ONLY IF

To ensure clarity, we will:

- Define the precise meaning of each connective
- Introduce its standard symbolic representation
- Look at the behavior of each operation through a truth table

Logical Conjunction (AND)

If p and q are propositions, their conjunction, "p and q", denoted by $p \wedge q$, is defined by the truth table:

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1

Here, note that:

- The symbols p, q, and r are placeholders for propositions
- Each row in the table represents one possible case
- 0 and 1 denote false and true, respectively

Finally, the conjunction $p \wedge q$ is true only when both p and q are true, just as in ordinary language.

Logical Conjunction (AND)

- Example

Suppose we want to check that $x > 0$ and $x < 5$.

This means x must satisfy both conditions simultaneously:

$$(x > 0) \land (x < 5)$$

The truth table below shows how the conjunction evaluates for different values of x:

x	$x > 0$	$x < 5$	$(x > 0) \land (x < 5)$
0	0	1	0
3	1	1	1
5	1	0	0

Logical Conjunction (AND)

- Example (Python)

In python we can write this as:

```
x = 3                               # Example value
if x > 0 and x < 5:                 # Conditional statement
    print("x is between 0 and 5")   # Print a message as the check passed
```

Logical Disjunction (OR)

If p and q are propositions, their disjunction, "p or q", denoted by $p \vee q$, is defined by:

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1

This operation reflects the inclusive or, meaning the result is true if either or both propositions are true.

Logical Disjunction (OR)

- Example

A quadratic equation $x^2 = 9$ has two possible solutions:

$$(x = 3) \lor (x = -3)$$

The truth table below shows how the disjunction evaluates for different values of x:

x	$x = 3$	$x = -3$	$(x = 3) \lor (x = -3)$
-3	0	1	1
0	0	0	0
3	1	0	1

Logical Disjunction (OR)

- Example (Python)

In python we can write this as:

```python
x = -3                                  # Example value
if x == 3 or x == -3:                   # Conditional statement
    print("x is a solution to x^2 = 9")  # Print a message as the check passed
```

Logical Negation (NOT)

Negation, denoted by $\neg p$, is the only standard operation that applies to a single proposition.

p	$\neg p$
0	1
1	0

Logical Negation (NOT)

- Example

To express that x is not equal to 5:

$$\neg(x = 5)$$

The truth table below shows how the negation evaluates for different values of x:

x	$x = 5$	$\neg(x = 5)$
5	1	0
7	0	1
0	0	1

Logical Negation (NOT)

- Example (Python)

In python we can write this as:

```
x = 7                    # Example value
if not x == 5:           # Conditional statement
    print("x is not 5")  # Print a message as the check passed
```

Conditional Statement (IF...THEN...)

The conditional statement “If p then q”, denoted $p \to q$, is defined by:

p	q	$p \to q$
0	0	1
0	1	1
1	0	0
1	1	1

The conditional is false only when p is true and q is false.

Conditional Statement (IF...THEN...)

- Example

Consider the statement "If a number is greater than 10, then it is even", which we can write more formally as:

$$(x > 2) \rightarrow (x \text{ is even })$$

The truth table below shows how the conditional evaluates for different values of x:

x	$x > 10$	x is even	$(x > 2) \rightarrow (x \text{ is even })$
12	1	1	1
11	1	0	0
8	0	1	1
7	0	0	1

Conditional Statement (IF...THEN...)

- Example (Python)

In python we can write this as:

```
1   x = 11                    # Example value
2   if x > 10 and x % 2 != 0: # Conditional statement
3       print("Success!")     # Print a message as the check passed
```

Biconditional (IF AND ONLY IF)

If p and q are propositions, the biconditional, "p if and only if q", denoted $p \leftrightarrow q$, is defined by:

p	q	$p \leftrightarrow q$
0	0	1
0	1	0
1	0	0
1	1	1

The biconditional:

- is true when p and q share the same truth value, i.e., both true or both false
- is the same as saying p and q are logically equivalent, written as $p \equiv q$

Biconditional (IF AND ONLY IF)

- Example

The condition "x is not less than 5" is equivalent to "x is greater than or equal to 5":

$$\neg(x < 5) \leftrightarrow x \geq 5$$

The truth table below shows how the equivalence evaluates for different values of x:

x	$x < 5$	$\neg(x < 5)$	$x \geq 5$	$\neg(x < 5) \leftrightarrow x \geq 5$
3	1	0	0	1
5	0	1	1	1
7	0	1	1	1

Biconditional (IF AND ONLY IF)

- Example (Python)

In python we can write this as:

```
1   x = 7                   # Example value
2   if not x < 5 == x >= 5:  # Conditional statement
3       print("Success!")    # Print a message as the check passed
```

Tautology

A tautology is a logical expression that is true in every possible case.

The symbol 1 is often used to denote a tautology.

Tautology

- Example

The statement:

$$(x = 5) \vee (x \neq 5)$$

is a tautology, because it is true for any value of x.

The truth table below illustrates this:

x	$x = 5$	$x \neq 5$	$(x = 5) \vee (x \neq 5)$
5	1	0	1
3	0	1	1
42	0	1	1

Tautology

- Example (Python)

In python we can write this as:

```python
x = 5                              # Example value
if x == x or x != x:               # Conditional statement
    print("This is always true")  # Print a message as the check passed
```

Contradiction

A contradiction is a logical expression that is false in every possible case.

The symbol 0 is often used to denote a contradiction.

Contradiction

- Example

The statement:

$$(x > 5) \wedge (x < 5)$$

is a contradiction because it can never be true for any value of x.

The truth table below illustrates this:

x	$x > 5$	$x < 5$	$(x > 5) \wedge (x < 5)$
3	0	1	0
5	0	0	0
7	1	0	0

Contradiction

- Example (Python)

In python we can write this as:

```python
x = 5                              # Example value
if x > 5 and x < 5:                # Conditional statement
    print("This will never run")  # Print a message as the check passed
```

Exercise Set

Construct the truth tables for the following logical expressions:

1. $p \vee (\neg p)$
2. $p \wedge (\neg p)$

For each scenario below:

- Identify the propositions (p, q, etc.)
- Write the compound proposition symbolically
- State if the proposition is true or false

3. The variable `country` is either `"US"` or `"Canada"`
4. If the variable `income` > 100000, then the label `high_value` is `True`

For each of the following (Python conditional statements), determine if it is a tautology, contradiction, or neither:

5. `if (x > 10) or (not (x > 10)):`
6. `if (x in S) and (x not in S):`
7. `if (x in S) or (y in S):`

Note: `S` represents a Python set. For example:

```
S = {1, 2, 3, 4} # Define a set
x = 1            # Number of interest
x in S           # Determine if x is in S
```