Introduction to logic

Benjamin Wack (benjamin.wack@univ-grenoble-alpes.fr)
Paolo Torrini (paolo.torrini@inria.fr)

Handout by Stéphane Devismes  Pascal Lafourcade  Michel Lévy

Université Grenoble Alpes

January 2021
Organization

12 weeks:

- Lecture, 1h30 / week
- Seminar 2 × 1h30 = 3h / week

This week

- Lecture: today! and Thursday at 15h15 as usual
- Seminar: 1 session on Wednesday this week, 2 starting next week

https://wackb.gricad-pages.univ-grenoble-alpes.fr/inf402/
## Final mark

### Evaluations

- **Assessments 60%:**
  - 4 periodic tests **10%**, midterm exam **20%** and project **30%**
- **Exam: 40%**

Project groups: 3-4 students per project group.

- **Part 1:** Modeling of a logic problem (automated in a software)
- **Part 2:** Transforming instances of these problems in clauses and solving them using an SAT solver

Examples of problems: N queens, Sudoku-like grids...
Planning

Important dates

- Project pre-report: March 5th
- Midterm exam: March 8th - 12th
- Project report: April 30th
- Project defense: May 3rd - May 12th
- Final exam: May 17th - 28th
- Second session: June 21th - 25th
Course Material

- Lectures handout (in French, with holes)
- Subject of the project (on the website)
Summary

Prerequisites

Introduction to Logic

Propositional Logic

Syntax

Meaning of formulae (a.k.a. Semantics)

Conclusion
Summary

Prerequisites

Introduction to Logic

Propositional Logic

Syntax

Meaning of formulae (a.k.a. Semantics)

Conclusion
Logic

Definitions

- Logic is used to specify what a correct reasoning is, regardless of the application domain.
Logic

Definitions

- Logic is used to specify what a correct reasoning is, regardless of the application domain.
- A reasoning is a way to obtain a conclusion starting from given hypotheses.
Logic

Definitions

- **Logic** is used to specify what a correct reasoning is, regardless of the application domain.

- A **reasoning** is a way to obtain a conclusion starting from given hypotheses.

- A **correct** reasoning does not say anything about the truth of the hypotheses, it only says that **starting from the truth of the hypotheses, one can deduct the truth of the conclusion.**
Examples

Example I

- **Hypothesis I:** All men are mortal
- **Hypothesis II:** Socrates is a man
- **Conclusion:** Socrates is mortal
Examples

Example I

- **Hypothesis I:** All men are mortal
- **Hypothesis II:** Socrates is a man
- **Conclusion:** Socrates is mortal

Example II

- **Hypothesis I:** All that is rare is expensive
- **Hypothesis II:** A cheap horse is rare
- **Conclusion:** A cheap horse is expensive!
Adding a hypothesis

Example III

Hypothesis I: All that is rare is expensive

Hypothesis II: A cheap horse is rare

Hypothesis III: Every cheap thing is "not expensive"

Conclusion: Contradictory hypotheses! Since:

Hypothesis I + Hypothesis II: A cheap horse is expensive

Hypothesis III: A cheap horse is not expensive
Adding a hypothesis

Example III

- **Hypothesis I**: All that is rare is expensive
- **Hypothesis II**: A cheap horse is rare
- **Hypothesis III**: Every cheap thing is “not expensive”
Adding a hypothesis

Example III

- **Hypothesis I:** All that is rare is expensive
- **Hypothesis II:** A cheap horse is rare
- **Hypothesis III:** Every cheap thing is “not expensive”
- **Conclusion:** Contradictory hypotheses! Since:
  - **Hypothesis I + Hypothesis II:** A cheap horse is expensive
  - **Hypothesis III:** A cheap horse is not expensive
Some history...

- **George Boole** (1815-1864)
  - *symbolic logic*: first try at reasoning without natural language

- **Gottlob Frege** (1848-1925)
  - *propositional calculus*: formal rules for reasoning
  - *proof theory*: a proof itself becomes a mathematical object

- **Bertrand Russell** (1872-1970)
  - *logicism*: attempt at a formalization of all existing mathematics
  - *paradox* found in earlier systems

- **Kurt Gödel** (1906-1978)
  - *completeness* of the first-order predicate calculus
  - *incompleteness theorem* for systems including arithmetic

- **Alonzo Church** (1903-1995)
  - *lambda-calculus*: a proof is an algorithm and vice-versa
Applications

- **Hardware**: logic gates
- **Software verification and correctness**: Tools: provers Coq, HOL, PVS, Prover9, MACE, ... Meteor (ligne 14)
- **Artificial Intelligence**: expert system (MyCin), ontology
- **Programming**: Prolog artificial intelligence natural language processing
- **Mathematical proofs, Security, ...**
Course Objectives

▶ Modeling and formalizing a problem.
▶ Understanding a formal reasoning, in particular, being able to determine if it is correct or not.
Course Objectives

▶ Modeling and formalizing a problem.
▶ Understanding a formal reasoning, in particular, being able to determine if it is correct or not.
▶ Reasoning, that is, building a correct reasoning using the tools of propositional logic and first order logic.
Course Objectives

- Modeling and formalizing a problem.
- Understanding a formal reasoning, in particular, being able to determine if it is correct or not.
- Reasoning, that is, building a correct reasoning using the tools of propositional logic and first order logic.
Course Objectives

- Modeling and formalizing a problem.
- Understanding a formal reasoning, in particular, being able to determine if it is correct or not.
- Reasoning, that is, building a correct reasoning using the tools of propositional logic and first order logic.
- Writing a rigorous proof, in particular an induction.
Overview of the Semester

TODAY

▶ Propositional logic
▶ Propositional resolution
▶ Natural deduction for propositional logic

MIDTERM EXAM

▶ First order logic
▶ Logical basis for automated proving
  ("first-order resolution")
▶ First-order natural deduction

EXAM
Summary

Prerequisites

Introduction to Logic

Propositional Logic

Syntax

Meaning of formulae (a.k.a. Semantics)

Conclusion
Propositional Logic

Definition

**Propositional logic** is the logic *without quantifiers*. The only logical operations used are:

- $\neg$ (negation),
- $\land$ (conjunction, also known as logical “and”),
- $\lor$ (disjunction, also known as logical “or”),
- $\Rightarrow$ (implication)
- $\Leftrightarrow$ (equivalence)
Example: **Formal reasoning**

**Hypotheses:**
- (H1): If Peter is old, then John is not the son of Peter
- (H2): If Peter is not old, then John is the son of Peter
- (H3): If John is Peter’s son then Mary is the sister of John

**Conclusion** (C): Mary is the sister of John, or Peter is old.
Example: **Formal reasoning**

**Hypotheses:**
- (H1): If Peter is old, then John is not the son of Peter
- (H2): If Peter is not old, then John is the son of Peter
- (H3): If John is Peter’s son then Mary is the sister of John

**Conclusion (C):** Mary is the sister of John, or Peter is old.

- $p$: ”Peter is old”
- $j$: ”John is the son of Peter”
- $m$: ”Mary is the sister of John”
Example: **Formal reasoning**

**Hypotheses** :
- (H1): If Peter is old, then John is not the son of Peter
- (H2): If Peter is not old, then John is the son of Peter
- (H3): If John is Peter’s son then Mary is the sister of John

**Conclusion** (C): Mary is the sister of John, or Peter is old.

- $p$: "Peter is old"
- $j$: "John is the son of Peter"
- $m$: "Mary is the sister of John"
- (H1): $p \Rightarrow \neg j$
- (H2): $\neg p \Rightarrow j$
- (H3): $j \Rightarrow m$
- (C): $m \lor p$
Example: Formal reasoning

Hypotheses:
▸ (H1): If Peter is old, then John is not the son of Peter
▸ (H2): If Peter is not old, then John is the son of Peter
▸ (H3): If John is Peter’s son then Mary is the sister of John

Conclusion (C): Mary is the sister of John, or Peter is old.

▸ p: "Peter is old"
▸ j: "John is the son of Peter"
▸ m: "Mary is the sister of John"

We prove that \( H1 \land H2 \land H3 \Rightarrow C \):

\[
(p \Rightarrow \neg j) \land (\neg p \Rightarrow j) \land (j \Rightarrow m) \Rightarrow m \lor p
\]

is true regardless of the truth value of propositions \( p, j, m \).
Summary

Prerequisites

Introduction to Logic

Propositional Logic

Syntax

Meaning of formulae (a.k.a. Semantics)

Conclusion
Vocabulary of the language

FORMAL:

- The constants: \( \top \) (true) and \( \bot \) (false)
- The variables: for example, \( x, y_1 \)
- The parentheses
- The connectives: \( \neg, \lor, \land, \Rightarrow, \Leftrightarrow \)

INFORMAL:
we use metavariables, e.g. \( A, B, a, p \) to represent formulas
Definition 1.1.1 (strict well-formed formula)

A strict formula is defined inductively as:

- $\top$ and $\bot$ are strict formulae.
- A variable (e.g. $x$) is a strict formula.
- If $A$ is a strict formula then $\neg A$ is a strict formula.
- If $A$ and $B$ are strict formulae, and $\circ$ is one of the following operations: $\lor$, $\land$, $\Rightarrow$, $\Leftrightarrow$, then $(A \circ B)$ is a strict formula.
(Strict) Formula

Definition 1.1.1 (strict well-formed formula)

A strict formula is defined inductively as:

- \( \top \) and \( \bot \) are strict formulae.
- A variable (e.g. \( x \)) is a strict formula.
- If \( A \) is a strict formula then \( \neg A \) is a strict formula.
- If \( A \) and \( B \) are strict formulae, and \( \circ \) is one of the following operations: \( \lor, \land, \rightarrow, \leftrightarrow \), then \( (A \circ B) \) is a strict formula.

Example 1.1.2

\((a \lor (\neg b \land c))\) is a strict formula, but not \( a \lor (\neg b \land c) \), nor \( (a \lor (\neg b) \land c)) \).
Example 1.1.3

The structure of the formula \((a \lor (\neg b \land c))\) is illustrated by the following tree:
Syntax Tree

Example 1.1.3

The structure of the formula \((a \lor (\neg b \land c))\) is illustrated by the following tree:
Example 1.1.3

The structure of the formula \((a \lor (\neg b \land c))\) is illustrated by the following tree:

\[
\begin{array}{c}
\lor \\
\downarrow \\
\downarrow \\
a \\
\end{array}
\]
Syntax Tree

Example 1.1.3

The structure of the formula \((a \lor (\neg b \land c))\) is illustrated by the following tree:
Example 1.1.3

The structure of the formula \( (a \lor (\neg b \land c)) \) is illustrated by the following tree:
Exercise

\[((p \land \neg(p \lor q)) \land \neg r)\]
Exercise

\(((p \land \neg (p \lor q)) \land \neg r)\)
Size of a formula

Definition 1.1.10

The size of a formula $A$, denoted $|A|$, is inductively defined as:

- $|\top| = 0$ and $|\bot| = 0$.
- If $A$ is a variable then $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.
Size of a formula

Definition 1.1.10

The size of a formula $A$, denoted $|A|$, is inductively defined as:

- $|\top| = 0$ and $|\bot| = 0$.
- If $A$ is a variable then $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.

Example 1.1.11

$|(a \lor (\neg b \land c))| = $
Size of a formula

Definition 1.1.10

The size of a formula $A$, denoted $|A|$, is inductively defined as:

- $|\top| = 0$ and $|\bot| = 0$.
- If $A$ is a variable then $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.

Example 1.1.11

$|(a \lor (\neg b \land c))| = 3$.
First result

Strict formulae decompose uniquely into their sub-formulae.

Theorem 1.1.13 (no syntactic ambiguity)

For every formula $A$, there is one and only one of the following cases:

- $A$ is a variable,
- $A$ is a constant,
- $A$ can be written in a unique manner as $\neg B$ where $B$ is a formula,
- $A$ can be written in a unique manner as $(B \circ C)$ where $B$ and $C$ are formulae.

This will allow us to:

- prove properties by cases
- perform structural induction on the formulae rather than induction on their size.
Prioritized formula (well-formed formula)

Definition 1.1.14

A prioritized formula is inductively defined in a similar way but:

- if $A$ and $B$ are prioritized formulae the $A \circ B$ is a prioritized formula,
- if $A$ is a prioritized formula then $(A)$ is a prioritized formula.

Example 1.1.15

$a \lor \neg b \land c$ is a prioritized formula, but not a (strict) formula.
Connective precedence

Definition 1.1.16
By decreasing precedence, the connectives are: ¬, ∧, ∨, ⇒ and ⇔.

Left associativity
For identical connectives, the left-hand side connective has higher precedence:
\[ A \circ B \circ C = (A \circ B) \circ C \]
except for the implication: \[ A \Rightarrow B \Rightarrow C = A \Rightarrow (B \Rightarrow C) \]
Example of prioritized formulae

Example 1.1.17

- $a \land b \land c$ is the abbreviation of
- $a \land b \lor c$ is the abbreviation of
- $a \lor b \land c$ is the abbreviation of
Example of prioritized formulae

Example 1.1.17

- $a \land b \land c$ is the abbreviation of $(a \land b) \land c$

- $a \land b \lor c$ is the abbreviation of $(a \land b) \lor c$

- $a \lor b \land c$ is the abbreviation of $(a \lor (b \land c))$
Summary

Prerequisites

Introduction to Logic

Propositional Logic

Syntax

Meaning of formulae (a.k.a. Semantics)

Conclusion
Definition 1.2.1

A truth assignment is a function from the set of variables of a formula to the set \( \{0, 1\} \).

\([A]_v\) denotes the truth value of the formula \( A \) for the assignment \( v \).
Truth assignment of a formula

Definition 1.2.1

A truth assignment is a function from the set of variables of a formula to the set \{0, 1\}.

\([A]_\nu\) denotes the truth value of the formula \(A\) for the assignment \(\nu\).

Example: Let \(\nu\) be an assignment such that \(\nu(x) = 0\) and \(\nu(y) = 1\).

Applying \(\nu\) to \(x \lor y\) is written as
Truth assignment of a formula

**Definition 1.2.1**

A truth assignment is a function from the set of variables of a formula to the set \( \{0, 1\} \).

\([A]_\nu\) denotes the truth value of the formula \( A \) for the assignment \( \nu \).

**Example:** Let \( \nu \) be an assignment such that \( \nu(x) = 0 \) and \( \nu(y) = 1 \).

Applying \( \nu \) to \( x \lor y \) is written as \([x \lor y]_\nu\)

\([x \lor y]_\nu = \)

Truth assignment of a formula

Definition 1.2.1
A truth assignment is a function from the set of variables of a formula to the set \{0, 1\}. \([A]_\nu\) denotes the truth value of the formula \(A\) for the assignment \(\nu\).

Example: Let \(\nu\) be an assignment such that \(\nu(x) = 0\) and \(\nu(y) = 1\). Applying \(\nu\) to \(x \lor y\) is written as \([x \lor y]_\nu\)
\([x \lor y]_\nu = 0 \lor 1 = 1\)
Conclusion:
Truth assignment of a formula

Definition 1.2.1
A truth assignment is a function from the set of variables of a formula to the set \( \{0, 1\} \).

\([A]_\nu\) denotes the truth value of the formula \( A \) for the assignment \( \nu \).

**Example:** Let \( \nu \) be an assignment such that \( \nu(x) = 0 \) and \( \nu(y) = 1 \).
Applying \( \nu \) to \( x \lor y \) is written as \([x \lor y]_\nu\)

\([x \lor y]_\nu = 0 \lor 1 = 1\)

Conclusion: \( x \lor y \) is true for the truth assignment \( \nu \)
Truth value of a formula

The value of $[A]_v$ is defined by structural induction on $A$, given the truth assignment $v$.

Definition 1.2.2 (truth interpretation of a formula)

- $[x]_v = \ldots$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = \ldots$
- $[(A \lor B)]_v = \ldots$
- $[(A \land B)]_v = \ldots$
- $[(A \Rightarrow B)]_v = \ldots$
- $[(A \Leftrightarrow B)]_v = \ldots$
Truth value of a formula

The value of \([A]_\nu\) is defined by structural induction on \(A\), given the truth assignment \(\nu\).

Definition 1.2.2 (truth interpretation of a formula)

- \([x]_\nu = \nu(x)\)
- \([\top]_\nu = 1, [\bot]_\nu = 0\)
- \([\neg A]_\nu = 1 - [A]_\nu\)
- \([(A \lor B)]_\nu = \max\{[A]_\nu, [B]_\nu\}\)
- \([(A \land B)]_\nu = \min\{[A]_\nu, [B]_\nu\}\)
- \([(A \Rightarrow B)]_\nu = \begin{cases} 1 & \text{if } [A]_\nu = 0 \\ [B]_\nu & \text{else} \end{cases}\)
- \([(A \iff B)]_\nu = \begin{cases} 1 & \text{if } [A]_\nu = [B]_\nu \\ 0 & \text{else} \end{cases}\)
Truth value of a formula

The value of $[A]_v$ is defined by structural induction on $A$, given the truth assignment $v$.

**Definition 1.2.2 (truth interpretation of a formula)**

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v =$
- $[(A \lor B)]_v =$
- $[(A \land B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$
Truth value of a formula

The value of $[A]_v$ is defined by structural induction on $A$, given the truth assignment $v$.

**Definition 1.2.2 (truth interpretation of a formula)**

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v =$
- $[(A \lor B)]_v =$
- $[(A \land B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \iff B)]_v =$
Truth value of a formula

The value of $[A]_\nu$ is defined by structural induction on $A$, given the truth assignment $\nu$.

**Definition 1.2.2 (truth interpretation of a formula)**

- $[x]_\nu = \nu(x)$
- $[\top]_\nu = 1$, $[\bot]_\nu = 0$
- $[\neg A]_\nu = 1 - [A]_\nu$
- $[(A \lor B)]_\nu = \max\{[A]_\nu, [B]_\nu\}$
- $[(A \land B)]_\nu = \min\{[A]_\nu, [B]_\nu\}$
- $[(A \Rightarrow B)]_\nu = \begin{cases} 1 & \text{if } [A]_\nu = 0 \\ [B]_\nu & \text{otherwise} \end{cases}$
- $[(A \Leftrightarrow B)]_\nu = \begin{cases} 1 & \text{if } [A]_\nu = [B]_\nu \\ 0 & \text{otherwise} \end{cases}$
Truth value of a formula

The value of $[A]_v$ is defined by structural induction on $A$, given the truth assignment $v$.

**Definition 1.2.2 (truth interpretation of a formula)**

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \lor B)]_v = \max\{[A]_v, [B]_v\}$
- $[(A \land B)]_v = \min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v = \begin{cases} 1 & \text{if } [A]_v = 0 \\ [B]_v & \text{otherwise} \end{cases}$
- $[(A \Leftrightarrow B)]_v = \begin{cases} 1 & \text{if } [A]_v = [B]_v \\ 0 & \text{otherwise} \end{cases}$
Truth value of a formula

The value of $[A]_\nu$ is defined by structural induction on $A$, given the truth assignment $\nu$.

Definition 1.2.2 (truth interpretation of a formula)

- $[x]_\nu = \nu(x)$
- $[\top]_\nu = 1$, $[\bot]_\nu = 0$
- $[\neg A]_\nu = 1 - [A]_\nu$
- $[(A \lor B)]_\nu = \max\{[A]_\nu, [B]_\nu\}$
- $[(A \land B)]_\nu = \min\{[A]_\nu, [B]_\nu\}$
- $[(A \Rightarrow B)]_\nu =$
- $[(A \Leftrightarrow B)]_\nu =$
Truth value of a formula

The value of \([A]_v\) is defined by structural induction on \(A\), given the truth assignment \(v\).

Definition 1.2.2 (truth interpretation of a formula)

- \([x]_v = v(x)\)
- \([\top]_v = 1, [\bot]_v = 0\)
- \([\neg A]_v = 1 - [A]_v\)
- \([A \lor B]_v = \max\{[A]_v, [B]_v\}\)
- \([A \land B]_v = \min\{[A]_v, [B]_v\}\)
- \([A \Rightarrow B]_v = \text{if } [A]_v = 0 \text{ then } 1 \text{ else } [B]_v\)
- \([A \iff B]_v = \)
Truth value of a formula

The value of $[A]_\nu$ is defined by structural induction on $A$, given the truth assignment $\nu$.

Definition 1.2.2 (truth interpretation of a formula)

- $[x]_\nu = \nu(x)$
- $[\top]_\nu = 1$, $[\bot]_\nu = 0$
- $[\neg A]_\nu = 1 - [A]_\nu$
- $[(A \lor B)]_\nu = \max\{[A]_\nu, [B]_\nu\}$
- $[(A \land B)]_\nu = \min\{[A]_\nu, [B]_\nu\}$
- $[(A \Rightarrow B)]_\nu = \text{if } [A]_\nu = 0 \text{ then } 1 \text{ else } [B]_\nu$
- $[(A \Leftrightarrow B)]_\nu = \text{if } [A]_\nu = [B]_\nu \text{ then } 1 \text{ else } 0$
Truth table

Definition 1.2.3

A truth table of a formula $A$ is a table representing the truth values of $A$ for all the possible values of the variables of $A$.

- a line of the truth table = an assignment
- a column of the truth table = the truth values of a formula.
Basic truth tables

0 indicates false and 1 indicates true.
The value of the constant $\top$ is 1 and the value of the constant $\bot$ is 0

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$\neg x$</th>
<th>$x \lor y$</th>
<th>$x \land y$</th>
<th>$x \Rightarrow y$</th>
<th>$x \Leftrightarrow y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Example 1.2.4

Give the truth table of the following formulae.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \implies y$</th>
<th>$\neg x$</th>
<th>$\neg x \lor y$</th>
<th>$(x \implies y) \Leftrightarrow (\neg x \lor y)$</th>
<th>$x \lor \neg y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 1.2.4

Give the truth table of the following formulae.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \Rightarrow y$</th>
<th>$\neg x$</th>
<th>$\neg x \lor y$</th>
<th>$(x \Rightarrow y) \Leftrightarrow (\neg x \lor y)$</th>
<th>$x \lor \neg y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example:

Example 1.2.4

Give the truth table of the following formulae.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>x ⇒ y</th>
<th>¬x</th>
<th>¬x ∨ y</th>
<th>(x ⇒ y) ⇔ (¬x ∨ y)</th>
<th>x ∨ ¬y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example 1.2.4

Give the truth table of the following formulae.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>x ⇒ y</th>
<th>¬x</th>
<th>¬x ∨ y</th>
<th>(x ⇒ y) ⇔ (¬x ∨ y)</th>
<th>x ∨ ¬y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Example:

**Example 1.2.4**

Give the truth table of the following formulae.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \Rightarrow y$</th>
<th>$\neg x$</th>
<th>$\neg x \lor y$</th>
<th>$(x \Rightarrow y) \Leftrightarrow (\neg x \lor y)$</th>
<th>$x \lor \neg y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>
Example:

Example 1.2.4

Give the truth table of the following formulae.

<table>
<thead>
<tr>
<th>$x$</th>
<th>$y$</th>
<th>$x \Rightarrow y$</th>
<th>$\neg x$</th>
<th>$\neg x \lor y$</th>
<th>$(x \Rightarrow y) \iff (\neg x \lor y)$</th>
<th>$x \lor \neg y$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Equivalent formulae

Definition 1.2.5

Two formulae $A$ and $B$ are equivalent (denoted $A \equiv B$ or simply $A = B$) if they have the same truth value for every assignment.
Equivalent formulae

Definition 1.2.5

Two formulae $A$ and $B$ are equivalent (denoted $A \equiv B$ or simply $A = B$) if they have the same truth value for every assignment.

Example 1.2.6

$$x \Rightarrow y \equiv \neg x \lor y$$
Equivalent formulae

Definition 1.2.5

Two formulae $A$ and $B$ are equivalent (denoted $A \equiv B$ or simply $A = B$) if they have the same truth value for every assignment.

Example 1.2.6

$x \Rightarrow y \equiv \neg x \lor y$

Remark:
The logical connective $\Leftrightarrow$ does not mean $A \equiv B$. 
Validity, tautology (1/2)

Definition 1.2.8

- A formula is valid if its value is 1 for all truth assignments.
- A valid formula is also called a tautology.
- Denoted by $\models A$. 

Example 1.2.9

$(x \Rightarrow y) \iff (\neg x \lor y)$ is valid; $x \Rightarrow y$ is not valid since it is false for $x = 1$ and $y = 0$. 
Validity, tautology (1/2)

Definition 1.2.8

- A formula is \textit{valid} if its value is 1 for all truth assignments.
- A valid formula is also called a \textit{tautology}.
- Denoted by $\models A$.

Example 1.2.9

- $(x \Rightarrow y) \iff (\neg x \lor y)$ is valid;
- $x \Rightarrow y$ is not valid since it is false for $x = 1$ and $y = 0$. 
Valid, tautology (2/2)

Property 1.2.10

The formulae $A$ and $B$ are equivalent ($A \equiv B$) if and only if formula $A \iff B$ is valid.

Proof.

The property is a consequence of the truth table of $\iff$. 

□
Model for a formula

Definition 1.2.11

A truth assignment $v$ for which a formula has truth value equal to 1 is a model for that formula.

$v$ satisfies $A$ or $v$ makes $A$ true.

Example 1.2.12

A model for $x \Rightarrow y$ is:

$x = 1, y = 1$ (among others)

On the opposite, $x = 1, y = 0$ is not a model for $x \Rightarrow y$. 


Model for a formula

Definition 1.2.11

A truth assignment $\nu$ for which a formula has truth value equal to 1 is a model for that formula.

$\nu$ satisfies $A$ or $\nu$ makes $A$ true.

Example 1.2.12

A model for $x \Rightarrow y$ is:

$x = 1, y = 1$ (among others)
Definition 1.2.11

A truth assignment $v$ for which a formula has truth value equal to 1 is a model for that formula.

$v$ satisfies $A$ or $v$ makes $A$ true.

Example 1.2.12

A model for $x \Rightarrow y$ is:

$x = 1, y = 1$ (among others)

On the opposite, $x = 1, y = 0$ is not a model for $x \Rightarrow y$. 
Model for a set of formulae

Definition 1.2.13

\( \nu \) is a model for a set of formulae \( \{ A_1, \ldots, A_n \} \)
if and only if
it is a model for every formula in the set.
Model for a set of formulae

Definition 1.2.13

\( \nu \) is a model for a set of formulae \( \{A_1, \ldots, A_n\} \)
if and only if
it is a model for every formula in the set.

Example 1.2.14

A model of \( \{a \Rightarrow b, b \Rightarrow c\} \) is:
Model for a set of formulae

Definition 1.2.13

$v$ is a model for a set of formulae $\{A_1, \ldots, A_n\}$ if and only if it is a model for every formula in the set.

Example 1.2.14

A model of $\{a \Rightarrow b, b \Rightarrow c\}$ is:

$a = 0, b = 0$ (for any $c$).
Property of a model for a set of formulae

Property 1.2.15

\( \nu \) is a model for \( \{ A_1, \ldots, A_n \} \)

if and only if

\( \nu \) is a model for \( A_1 \wedge \ldots \wedge A_n \).
Property of a model for a set of formulae

**Property 1.2.15**

$v$ is a model for $\{A_1, \ldots, A_n\}$ if and only if $v$ is a model for $A_1 \land \ldots \land A_n$.

**Example 1.2.16**

The set of formulae $\{a \Rightarrow b, b \Rightarrow c\}$ and the formula $(a \Rightarrow b) \land (b \Rightarrow c)$ have identical models.
Counter-model

Definition 1.2.17

A truth assignment $\nu$ which yields the value 0 for a formula is a counter-model for the formula.

$\nu$ does not satisfy the formula or $\nu$ makes the formula false.
Counter-model

Definition 1.2.17

A truth assignment $\nu$ which yields the value 0 for a formula is a counter-model for the formula.

$\nu$ does not satisfy the formula or $\nu$ makes the formula false.

Example 1.2.18

A counter-model of $x \Rightarrow y$ is:
Counter-model

Definition 1.2.17
A truth assignment $\nu$ which yields the value 0 for a formula is a counter-model for the formula.

$\nu$ does not satisfy the formula or $\nu$ makes the formula false.

Example 1.2.18
A counter-model of $x \Rightarrow y$ is:

$x = 1, y = 0.$
### Satisfiable formula

<table>
<thead>
<tr>
<th>Definition 1.2.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (set of) formula(e) is <strong>satisfiable</strong> if it admits a model.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Definition 1.2.21</th>
</tr>
</thead>
<tbody>
<tr>
<td>A (set of) formula(e) is <strong>unsatisfiable</strong> if it is not satisfiable.</td>
</tr>
</tbody>
</table>
Satisfiable formula

Definition 1.2.20
A (set of) formula(e) is **satisfiable** if it admits a model.

Definition 1.2.21
A (set of) formula(e) is **unsatisfiable** if it is not satisfiable.

Example 1.2.22

\[ x \land \neg x \text{ is unsatisfiable, but } x \Rightarrow y \text{ is satisfiable.} \]
Satisfiable formula

Definition 1.2.20
A (set of) formula(e) is satisfiable if it admits a model.

Definition 1.2.21
A (set of) formula(e) is unsatisfiable if it is not satisfiable.

Example 1.2.22
$x \land \neg x$ is unsatisfiable, but $x \implies y$ is satisfiable.

Beware
unsatisfiable = 0 model  
invalid = at least 1 counter-model

satisfiable = at least 1 model  
valid = 0 counter-model
Summary

Prerequisites

Introduction to Logic

Propositional Logic

Syntax

Meaning of formulae (a.k.a. Semantics)

Conclusion
Today

- Why define and use formal logic?
- Propositional logic:
  - 1 variable = 1 proposition (a statement) which may be true or false
  - 5 connectives to articulate these propositions
- Meaning of formulae:
  - assignment = choice of a truth value for each variable
  - a formula may be true for 0, 1, several or every assignment
Next time

Homework: build the truth table for the “Peter, John and Mary” example.

- Important equivalences
- Substitutions and replacements
- Normal Forms
Oxford’s motto

The more I study, the more I know
The more I know, the more I forget
The more I forget, the less I know