Introduction to logic

Stéphane Devismes   Pascal Lafourcade   Michel Lévy
Course given by Jean-François Monin
(jean-francois.monin@imag.fr)

Université Joseph Fourier, Grenoble I

January 16, 2015

Organisation

12 weeks :

- Course, 1h30 / week
- Seminar $2 \times 1h30 = 3h$ / week

Material :

- Course support (French : course notes (with holes))
- Subject of the project

Planning

Important dates

- *Winter break :* 1 week in February
- Midterm exam : following week
- *Spring break :* 1 week in April
- Project defense : end of April
- Final exam : relevant week in May
- Second session : relevant week in June

Final mark

Evaluations

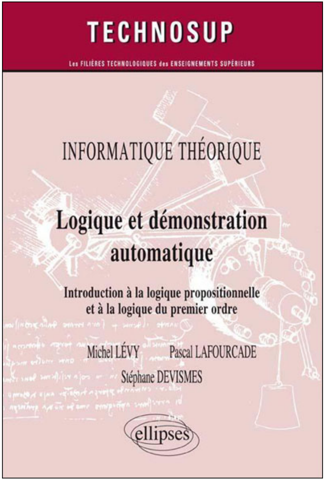
- Assessment **40%** : 4 periodic tests **10%**, midterm exam **40%** and project **50%**
- Exam : **60%**

Project groups : 3-4 students per project group.

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

Examples of problems : Squaro, Sudoku, Master Mind ...

Bibliography



Summary

Summary

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

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

Little history…

- **George Boole** (1815-1864), creator of modern logic (especially Boolean Algebra)
- **Friedrich Ludwig Gottlob Frege** (1848-1925), work on the modern propositional calculus, predicate calculus, proof theory
- **Bertrand Arthur William Russell** (1872-1970), application of logic to the question of the foundation of mathematics (logicism)
- **Alonzo Church** (1903-1995), *lambda-calculus*
- **Kurt GÃ¶del** (1906-1978), the *GÃ¶del's incompleteness theorems*, completeness of the first-order predicate calculus
- **Alan Mathison Turing** (1912-1954), father of computer science and artificial intelligence

Applications

- **Hardware** : The Arithmetic Logic Unit (ALU) is constructed from ≪ logic gates ≫
- **Software verification and correctness** :
  - Meteor (ligne 14)
  - Tools : provers COQ, PVS, Prover9, MACE, . . .
- **Artificial Intelligence** :
  - Turing Test
  - Decision making tool : expert system (*MyCin*), ontology
  - Semantic Web
- **SAT Problem** :
  - Coding a decision making problem as a Boolean expression
  - Applications in planning, model checking, diagnostic, . . .
  - Solvers : zchaff, satz, . . .
- **Programming** : Prolog is used by numerous artificial intelligence programs and for computer aided linguistic processing
- **Mathematical proofs**, **Security**, **. . .**

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

Course Objectives

- Understanding a reasoning, in particular, being able to determine if a logical reasoning is correct or not.

Course Objectives

- Understanding a reasoning, in particular, being able to determine if a logical reasoning is correct or not.
- Reasoning, that is, building a correct reasoning using the tools of propositional logic and first order logic.

Course Objectives

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

Course Objectives

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

Summary

Propositional Logic

Definition

Propositional logic is the logic *without quantifiers* which only uses the laws governing the following logical operations :

- $\neg$ (negation),
- $\wedge$ (conjunction, also known as logical “and”),
- $\vee$ (disjunction, also known as logical “or”),
- $\Rightarrow$ (implication) and
- $\Leftrightarrow$ (equivalence).

Remark

We limit our study to classical logic, which is the logic of two truth values : TRUE and FALSE

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"

- (H1) : $p \Rightarrow \neg j$
- (H2) : $\neg p \Rightarrow j$
- (H3) : $j \Rightarrow m$

(C) : $m \vee p$

We prove that $H1 \wedge H2 \wedge H3 \Rightarrow C$:

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

is true regardless of the truth value of the propositions p, j, m.

Summary

Vocabulary of the language

- The constants : $\top$ and $\bot$ representing *true* and *false* respectively.
- The variables : a variable is an identifier, with or without index, for example, x, y_1.
- The parentheses : left (and right).
- The connectives : $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$ respectively called negation, disjunction (or), conjunction (and), implication and equivalence.

(Strict) Formula

Definition 1.1.1

A strict formula is defined inductively as :

- $\top$ and $\perp$ are strict formulae.
- A variable 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 if $\circ$ is one of the following operations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ then $(A \circ B)$ is a strict formula.

(Strict) Formula

Definition 1.1.1

A strict formula is defined inductively as :

- $\top$ and $\bot$ are strict formulae.
- A variable 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 if $\circ$ is one of the following operations $\vee, \wedge, \Rightarrow, \Leftrightarrow$ then $(A \circ B)$ is a strict formula.

Example 1.1.2

$(a \vee (\neg b \wedge c))$ is a strict formula, but not $a \vee (\neg b \wedge c)$, nor $(a \vee (\neg(b) \wedge c))$.

Height of a formula

Definition 1.1.10

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

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

Height of a formula

Definition 1.1.10

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

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

Example 1.1.11

$|(a \vee (\neg b \wedge c))| =$

Height of a formula

Definition 1.1.10

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

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

Example 1.1.11

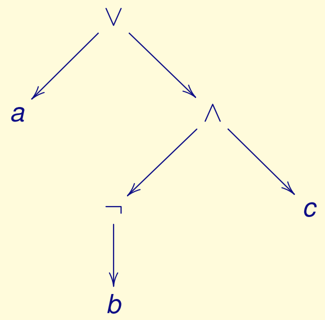
$|(a \vee (\neg b \wedge c))| =$

3.

Tree

Example 1.1.3

The structure of the following formula $(a \vee (\neg b \wedge c))$ is illustrated by the following tree :

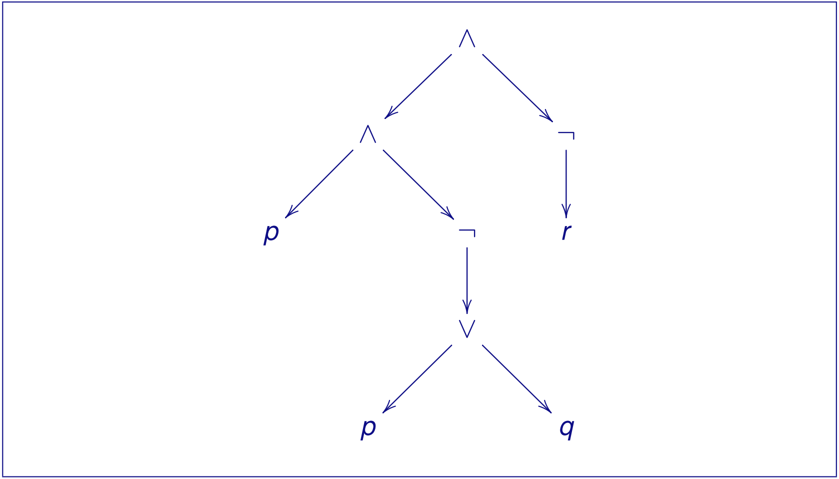


Exercise

$$((p \land \neg(p \lor q)) \land \neg r)$$

Exercise

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



Sub-formula

Definition 1.1.4

We call sub-formula of a (strict) formula A every factor of A which is a (strict) formula.

Example 1.1.5

$(\neg b \wedge c)$ is a sub-formula of $(a \vee (\neg b \wedge c))$.

A sub-formula of the formula A could be identified as a sub-tree of the tree representing the formula A.

First result

Strict formulae decompose uniquely in their sub-formulae

Theorem 1.1.13

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.

Proof.

Simple but tedious proof (*cf.* Course support) $\square$

Prioritized formula

Definition 1.1.14

A prioritized formula is inductively defined as :

- $\top$ and $\perp$ are prioritized formulae,
- a variable is a prioritized formula,
- if A is a prioritized formula then $\neg A$ is a prioritized formula,
- 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

$a \vee \neg b \wedge c$ is a prioritized formula, but not a (strict) formula.

Binding priorities

Definition 1.1.16

The decreasing order of binding priorities is as follows : $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$.

For equal priority, the left-hand side connective binds more tightly, **except for the implication** (which is right-associative).

Binding priorities

Definition 1.1.16

The decreasing order of binding priorities is as follows : $\neg$, $\wedge$, $\vee$, $\Rightarrow$ and $\Leftrightarrow$.

For equal priority, the left-hand side connective binds more tightly, **except for the implication** (which is right-associative).

A prioritized formula is the abbreviation of the (strict) associated formula.

Example of prioritized formula

Example 1.1.17

- $a \wedge b \wedge c$ is the abbreviation of
- $a \wedge b \vee c$ is the abbreviation of
- $a \vee b \wedge c$ is the abbreviation of

Example of prioritized formula

Example 1.1.17

- $a \wedge b \wedge c$ is the abbreviation of

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

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

  $(a \vee (b \wedge c))$

Summary

Basic tables

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

Table 1.1 (truth table of connectives)

x	y	$\neg x$	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

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 $\{0, 1\}$. Let A be a formula and v a truth assignment, $[A]_v$ denotes the truth value of the formula A for the truth 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 $\{0, 1\}$. Let A be a formula and v a truth assignment, $[A]_v$ denotes the truth value of the formula A for the truth assignment v.

Example : Let v a truth assignment such as $v(x) = 0$ and $v(y) = 1$

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 $\{0, 1\}$. Let A be a formula and v a truth assignment, $[A]_v$ denotes the truth value of the formula A for the truth assignment v.

Example : Let v a truth assignment such as $v(x) = 0$ and $v(y) = 1$
Applying the truth assignment v to $x \vee y$ is written as $[x \vee y]_v$

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 $\{0, 1\}$. Let A be a formula and v a truth assignment, $[A]_v$ denotes the truth value of the formula A for the truth assignment v.

Example : Let v a truth assignment such as $v(x) = 0$ and $v(y) = 1$
Applying the truth assignment v to $x \vee y$ is written as $[x \vee y]_v$
This equals $0 \vee 1 = 1$

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 $\{0, 1\}$. Let A be a formula and v a truth assignment, $[A]_v$ denotes the truth value of the formula A for the truth assignment v.

Example : Let v a truth assignment such as $v(x) = 0$ and $v(y) = 1$
Applying the truth assignment v to $x \vee y$ is written as $[x \vee y]_v$
This equals $0 \vee 1 = 1$
Conclusion : $x \vee y$ is true for the truth assignment v

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v =$
- $[\top]_v =\ \ $, $[\bot]_v =$
- $[\neg A]_v =$
- $[(A \vee B)]_v =$
- $[(A \wedge B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v =$, $[\bot]_v =$
- $[\neg A]_v =$
- $[(A \vee B)]_v =$
- $[(A \wedge B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v =$
- $[\neg A]_v =$
- $[(A \vee B)]_v =$
- $[(A \wedge B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v =$
- $[(A \vee B)]_v =$
- $[(A \wedge B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v =$
- $[(A \wedge B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v =$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v = min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v =$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v = min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v =$ if $[A]_v = 0$ then 1 else $[B]_v$
- $[(A \Leftrightarrow B)]_v =$

Truth value of a formula

Definition 1.2.2

Let A, B be two formulae, x a variable and v a truth assignment.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v = min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v =$ if $[A]_v = 0$ then 1 else $[B]_v$
- $[(A \Leftrightarrow B)]_v =$ if $[A]_v = [B]_v$ then 1 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 value of a formula.

Example :

Example 1.2.4

Give the truth table of the following formulae.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0					
0	1					
1	0					
1	1					

Example :

Example 1.2.4

Give the truth table of the following formulae.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1				
0	1	1				
1	0	0				
1	1	1				

Example :

Example 1.2.4

Give the truth table of the following formulae.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1			
0	1	1	1			
1	0	0	0			
1	1	1	0			

Example :

Example 1.2.4

Give the truth table of the following formulae.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1		
0	1	1	1	1		
1	0	0	0	0		
1	1	1	0	1		

Example :

Example 1.2.4

Give the truth table of the following formulae.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	
0	1	1	1	1	1	
1	0	0	0	0	1	
1	1	1	0	1	1	

Example :

Example 1.2.4

Give the truth table of the following formulae.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	1

Another example :

Give the truth table of

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0			
0	0	1			
0	1	0			
0	1	1			
1	0	0			
1	0	1			
1	1	0			
1	1	1			

Another example :

Give the truth table of

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1		
0	0	1	1		
0	1	0	0		
0	1	1	0		
1	0	0	1		
1	0	1	1		
1	1	0	0		
1	1	1	0		

Another example :

Give the truth table of

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1	0	
0	0	1	1	1	
0	1	0	0	0	
0	1	1	0	0	
1	0	0	1	0	
1	0	1	1	1	
1	1	0	0	0	
1	1	1	0	0	

Another example :

Give the truth table of

a	b	c	$\neg b$	$(\neg b \wedge c)$	$(a \vee (\neg b \wedge c))$
0	0	0	1	0	0
0	0	1	1	1	1
0	1	0	0	0	0
0	1	1	0	0	0
1	0	0	1	0	1
1	0	1	1	1	1
1	1	0	0	0	1
1	1	1	0	0	1

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

$x \Rightarrow y = \neg x \vee y$

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

$x \Rightarrow y = \neg x \vee y$

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.
- The fact that A is valid is denoted by $\models A$.

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.
- The fact that A is valid is denoted by $\models A$.

Example 1.2.9

- the formula $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ is valid ;
- the formula $x \Rightarrow y$ is not valid since

  it is false for the truth assignment $x = 1$ and $y = 0$,
  therefore it is not a tautology.

Valid, tautology (2/2)

Property 1.2.10

The formulae A and B are equivalent if and only if formula $A \Leftrightarrow B$ is valid.

Proof.

The property is a consequence of table 1.1 and of the previous definitions. $\square$

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.

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 :

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 :

$\boxed{x = 1, y = 1 \text{ where the truth assignment } x = 0 \text{ and any } y.}$

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 :

$x = 1, y = 1$ where the truth assignment $x = 0$ and any y.

On the contrary, the truth assignment $x = 1$ and $y = 0$ is not a model for $x \Rightarrow y$.

Model for a set of formulae

Definition 1.2.13

A truth assignment is a model for a set of formulae if and only if it is a model for every formula in the set.

Model for a set of formulae

Definition 1.2.13

A truth assignment is a model for a set of formulae 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

A truth assignment is a model for a set of formulae 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 :

$a = 0, b = 0$ (and for any c).

Property of a model for a set of formulae

Property 1.2.15

An assignment is a model for a set of formulae if and only if it is a model of the intersection of all the formulae in the set.

The proof is requested in the exercise 11.

Property of a model for a set of formulae

Property 1.2.15

An assignment is a model for a set of formulae if and only if it is a model of the intersection of all the formulae in the set.

The proof is requested in the exercise 11.

Example 1.2.16

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

Counter-model

Definition 1.2.17

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

v does not satisfy the formula or v makes the formula false.

Counter-model

Definition 1.2.17

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

v does not satisfy the formula or v 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 v which yields the value 0 for a formula is a counter-model for the formula.

v does not satisfy the formula or v makes the formula false.

Example 1.2.18

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

the assignment $x = 1, y = 0$.

Counter-model

Definition 1.2.17

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

v does not satisfy the formula or v makes the formula false.

Example 1.2.18

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

$\boxed{\text{the assignment } x = 1, y = 0.}$

Remark 1.2.19

The notion of counter-model applies to sets of formulae the same way as the notion of model.

Satisfiable formula

Definition 1.2.20

A formula (a set of formulae respectively) is satisfiable if there exists a truth assignment which is a model for the formula (or the set of formulae).

Definition 1.2.21

A formula (a set of formulae respectively) is unsatisfiable if it is not satisfiable.

Satisfiable formula

Definition 1.2.20

A formula (a set of formulae respectively) is satisfiable if there exists a truth assignment which is a model for the formula (or the set of formulae).

Definition 1.2.21

A formula (a set of formulae respectively) is unsatisfiable if it is not satisfiable.

Example 1.2.22

$x \wedge \neg x$ is unsatisfiable, but $x \Rightarrow y$ is not.

Satisfiable formula

Definition 1.2.20

A formula (a set of formulae respectively) is satisfiable if there exists a truth assignment which is a model for the formula (or the set of formulae).

Definition 1.2.21

A formula (a set of formulae respectively) is unsatisfiable if it is not satisfiable.

Example 1.2.22

$x \wedge \neg x$ is unsatisfiable, but $x \Rightarrow y$ is not.

Remark 1.2.23

Logicians use the term consistent as a synonym for satisfiable and contradictory as synonym of unsatisfiable.

Logical consequence (entailment)

Definition 1.2.24

A is a consequence of the set of hypotheses Γ ($\Gamma \models A$) if every model of Γ is model of A.

Remark 1.2.26

We denote by $\models A$ the fact that A is valid, since A is valid if and only if A is a consequence of the empty set.

Example of a consequence

Example 1.2.28

$a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c$.

Example of a consequence

Example 1.2.28

$a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c$.

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

ESSENTIAL property

Often used in exercises and during EXAMS.

Property 1.2.27

Let $A_1, \ldots, A_n$, B be $n+1$ formulae. Let H_n the conjunction of the formulae $A_1, \ldots, A_n$. The following three formulations are equivalent :

1. $A_1, \ldots, A_n \models B$, meaning that B is a consequence of the hypotheses $A_1, \ldots, A_n$.
2. The formula $H_n \Rightarrow B$ is valid.
3. $H_n \land \neg B$ is unsatisfiable.

Proof.

The property is a consequence of table 1.1 □

Proof (1/3)

- $1 \Rightarrow 2$: let us suppose that $A_1, \ldots, A_n \models B$: every model of $A_1, \ldots, A_n$ is also a model of B.
  - Let v be a truth assignment non-model of $A_1, \ldots, A_n : \exists i, [A_i]_v = 0$, therefore $[H_n]_v = 0$. Thus $[H_n \Rightarrow B]_v = 1$.
  - Let v be a truth assignment model of $A_1, \ldots, A_n$: $[A_i]_v = 1$ for $i = 1, \ldots, n$, therefore $[H_n]_v = 1$. But v is a model of B therefore $[B]_v = 1$. Thus $[H_n \Rightarrow B]_v = 1$.

Therefore $H_n \Rightarrow B$ is valid.

Proof (2/3)

- $2 \Rightarrow 3$: let us suppose that $H_n \Rightarrow B$ is valid : $\forall v$ truth assignment, $[H_n \Rightarrow B]_v = 1$.
  - let $[H_n]_v = 0$,
  - let $[H_n]_v = 1$ and $[B]_v = 1$.

  However $[H_n \wedge \neg B]_v = \min([H_n]_v, [\neg B]_v) = \min([H_n]_v, 1 - [B]_v)$. In both cases, we have $[H_n \wedge \neg B]_v = 0$. Therefore $H_n \wedge \neg B$ is unsatisfiable.

Proof (3/3)

- $3 \Rightarrow 1$: let us suppose that $H_n \wedge \neg B$ is unsatisfiable : $\forall$ truth assignment, the formula $H_n \wedge \neg B$ is contradictory.
  Let us show that the models of $A_1, \ldots, A_n$ are also models for B.
  Let v be a truth assignment model of $A_1, \ldots, A_n$: $[A_i]_v = 1$ for $i = 1, \ldots, n$ therefore $[H_n]_v = [A_1 \wedge \ldots \wedge A_n]_v = 1$.
  According to our hypothesis $[\neg B]_v = 0$. Hence, $1 - [B]_v = 0$. So $[B]_v = 1$, i.e. v is a model for B.

Using the result of exercise 7, we conclude.

Instance of the property

Example 1.2.28

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge \neg(a \Rightarrow c)$
0	0	0	1	1	1	1	0
0	0	1	1	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0
1	1	1	1	1	1	1	0

Compactness

Theorem 1.2.30 Propositional compactness

A set of propositional formulae has a model if an only if every finite subset of it has a model.

Compactness

Theorem 1.2.30 Propositional compactness

A set of propositional formulae has a model if an only if every finite subset of it has a model.

This theorem may look trivial. However, its proof is complex (*cf.* Poly). In order to understand the (difficulty of the) problem, it suffices to note that this theorem applies in particular to infinite sets of formulae ...

Compactness

Theorem 1.2.30 Propositional compactness

A set of propositional formulae has a model if an only if every finite subset of it has a model.

This theorem may look trivial. However, its proof is complex (*cf.* Poly). In order to understand the (difficulty of the) problem, it suffices to note that this theorem applies in particular to infinite sets of formulae ...

This result will be used at a later stage in the course (basis for the automated theorem proving).

Summary

Disjunction

Disjunction

- associative $x \vee (y \vee z) \equiv (x \vee y) \vee z$

Disjunction

- associative $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- commutative $x \vee y \equiv y \vee x$

Disjunction

- associative $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- commutative $x \vee y \equiv y \vee x$
- idempotent $x \vee x \equiv x$

Conjunction

Conjunction

- associative $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$

Conjunction

- associative $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$
- commutative $x \wedge y \equiv y \wedge x$

Conjunction

- associative $x \wedge (y \wedge z) \equiv (x \wedge y) \wedge z$
- commutative $x \wedge y \equiv y \wedge x$
- idempotent $x \wedge x \equiv x$

Distributivity

Distributivity

- Multiplication is distributive over addition
  $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$

Distributivity

- Multiplication is distributive over addition
  $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
- Addition is distributive over multiplication
  $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$

Neutrality and Absorption

Neutrality and Absorption

- 0 is a neutral element for disjunction $0 \vee x \equiv x$
- 1 is a neutral element for conjunction $1 \wedge x \equiv x$

Neutrality and Absorption

- 0 is a neutral element for disjunction $0 \vee x \equiv x$
- 1 is a neutral element for conjunction $1 \wedge x \equiv x$
- 1 is an absorbing element for disjunction $1 \vee x \equiv 1$
- 0 is an absorbing element for conjunction $0 \wedge x \equiv 0$

Negation

- Negation laws :
  - $x \wedge \neg x \equiv 0$.
  - $x \vee \neg x \equiv 1$ (The law of excluded middle).

Negation

- Negation laws :
  - $x \wedge \neg x \equiv 0$.
  - $x \vee \neg x \equiv 1$ (The law of excluded middle).
- $\neg\neg x \equiv x$.
- $\neg 0 \equiv 1$.
- $\neg 1 \equiv 0$.

De Morgan laws

- $\neg(x \wedge y) \equiv \neg x \vee \neg y$.
- $\neg(x \vee y) \equiv \neg x \wedge \neg y$.

Simplification laws

Property 1.2.31

For every x, y we have :

- $x \vee (x \wedge y) \equiv x$
- $x \wedge (x \vee y) \equiv x$
- $x \vee (\neg x \wedge y) \equiv x \vee y$

Proof.

The proof is requested in exercise 12. $\square$

Summary

Conclusion : Today

- Introduction and history
- Propositional logic
- Syntax
- Meaning of formulae
- Important Equivalences

Conclusion : Next course

- Substitutions and replacements
- Normal Forms
- Boolean Algebra
- Boolean functions
- The BDDC tool

Conclusion

Thank you for your attention.

Questions ?

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