Transformations of logical formulas

Benjamin Wack

Université Grenoble Alpes

January 2025

Previous lecture

- Why formal logic ?
- Propositional logic
- Syntax
- Meaning of formulas

Our example with a truth table

Hypotheses:

- (H1): If Peter is tall, then John is not the son of Peter
- (H2): If Peter is not tall, 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 tall.

$$(p \Rightarrow \neg j) \land (\neg p \Rightarrow j) \land (j \Rightarrow m) \Rightarrow m \lor p$$

p	j	m	$p \Rightarrow \neg j$	$\neg p \Rightarrow j$	$j \Rightarrow m$	$H_1 \land H_2 \land H_3$	$m \lor p$	$H_1 \land H_2 \land H_3 \Rightarrow m \lor p$
0	0	0	1	0	1	0	0	1
0	0	1	1	0	1	0	1	1
0	1	0	1	1	0	0	0	1
0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	1	0	0	1	1
1	1	1	0	1	1	0	1	1

Plan

Logical consequence (entailment)

Definition 1.2.24

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

Remark 1.2.26

$\models A$ denotes that A is valid.
(Every truth assignment is a model for the empty set.)

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 $H_n = A_1 \wedge \ldots \wedge A_n$.
The following three formulations are equivalent:

1. $A_1, \ldots, A_n \models B$
2. $H_n \Rightarrow B$ is valid.
3. $H_n \wedge \neg B$ is unsatisfiable.

Proof.

Based on the truth tables of the connectives.
We prove that $1 \Rightarrow 2$ then $2 \Rightarrow 3$ and $3 \Rightarrow 1$. $\square$

Proof (1/3)

- $1 \Rightarrow 2$: let us assume that $A_1, \ldots, A_n \models B$.

Let v be a truth assignment:

- if v is not a model for $A_1, \ldots, A_n$:
  for a certain i we have $[A_i]_v = 0$, hence $[H_n]_v = 0$.
- if v is a model for $A_1, \ldots, A_n$:
  then by hypothesis v is a model for B therefore $[B]_v = 1$.

In each case $[H_n \Rightarrow B]_v = 1$, therefore $H_n \Rightarrow B$ is valid.

Proof (2/3)

- $2 \Rightarrow 3$: let us assume that $H_n \Rightarrow B$ is valid.
  For every truth assignment v:
  - either $[H_n]_v = 0$,
  - or $[H_n]_v = 1$ 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 assume that $H_n \wedge \neg B$ is unsatisfiable.
  Let us show that $A_1, \ldots, A_n \models B$.
  Let v be a truth assignment model of $A_1, \ldots, A_n$:
  - $[H_n]_v = [A_1 \wedge \ldots \wedge A_n]_v = 1$.
  - 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.

Exercise 7 shows why proving these 3 circular implications is sufficient.

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

Preamble

How to prove that a formula is valid?

- Truth table
  - Problem: for a formula having 100 variables, the truth table will contain 2^{100} lines (unfeasible, even with a computer!).
- Idea:
  - Simplify the formula using transformations
  - Then, study the simplified formula using truth tables or a logic reasoning

Disjunction

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

Ditto for conjunction.

Distributivity

- Conjunction distributes over disjunction
  $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
- Disjunction distributes over conjunction
  $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$

Neutrality and Absorption

- $\perp$ is the neutral element for disjunction $\perp \vee x \equiv x$
- $\top$ is the neutral element for conjunction $\top \wedge x \equiv x$
- $\top$ is the absorbing element for disjunction $\top \vee x \equiv \top$
- $\perp$ is the absorbing element for conjunction $\perp \wedge x \equiv \perp$

Vercors is full of wonders

Where can you go if you are both “*exploitant*” and “*ayant droit*”?

Negation

- Negation laws:
  - $x \wedge \neg x \equiv \bot$
  - $x \vee \neg x \equiv \top$ (The law of excluded middle)
- $\neg\neg x \equiv x$
- $\neg\bot \equiv \top$
- $\neg\top \equiv \bot$

De Morgan's laws

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

Augustus De Morgan (1860) builds on Boole's algebra:

- Work about quantifiers
- Calculus of relations
  (also see C.S. Peirce's works)

which laid grounds for first-ordre logic (see 2nd part of the course).

- Notion of duality in Boole's algebras

expressed in particular as De Morgan's laws

- Involved (though very briefly) in the first conjectures about the four colour theorem

Simplification laws

Property 1.2.31

For every x, y we have:

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

Substitution

Definition 1.3.1

A substitution σ is a function mapping variables to formulas.

$\sigma(A)$ = the formula A where all variables x are replaced by the formula $\sigma(x)$.

Example: $A = \neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$

- Let σ the following substitution: $\sigma(p) = (a \vee b), \sigma(q) = (c \wedge d)$
- $\sigma(A) = \neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d))$

Finite domain substitution

Definition 1.3.2

- If a substitution changes only a finite number of variables we note it $< x_1 := A_1, \ldots, x_n := A_n >$

Example 1.3.3

$A = x \vee x \wedge y \Rightarrow z \wedge y$ and $\sigma =< x := a \vee b, z := b \wedge c >$

$\boxed{\sigma(A) = (a \vee b) \vee (a \vee b) \wedge y \Rightarrow (b \wedge c) \wedge y}$

Substitution of a valid formula

(Let us assume property 1.3.4.)

Theorem 1.3.6

If A is valid then $\sigma(A)$ is valid for any σ.

Example 1.3.7

Let A be the formula $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$.
This formula is valid, it is an important equivalence.
Let σ the following substitution: $< p := (a \vee b), q := (c \wedge d) >$.
The formula

$\boxed{\sigma(A) = \neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d)) \text{ is also valid.}}$

Consequence : important equivalences can be applied to any formulas, not just to variables.

Replacement

Definition 1.3.8

The formula D is obtained by replacing certain occurrences of A by B in C if:

- C can be written $E < x := A >$
- D can be written $E < x := B >$

for some formula E.

Examples

Example 1.3.9

Consider the formula $C = ((a \Rightarrow b) \vee \neg(a \Rightarrow b))$.

- The formula obtained by replacing all occurrences of $(a \Rightarrow b)$ by $(a \wedge b)$ is

  $$\boxed{D = ((a \wedge b) \vee \neg(a \wedge b))}$$

  using $E = (x \vee \neg x)$.
- The formula obtained by replacing the *first* occurrence of $(a \Rightarrow b)$ by $(a \wedge b)$ is

  $$\boxed{D = ((a \wedge b) \vee \neg(a \Rightarrow b))}$$

  using $E = (x \vee \neg(a \Rightarrow b))$.

Properties of the replacements

Theorem 1.3.10

If by replacing A by B inside C we obtain D, then $(A \Leftrightarrow B) \models (C \Leftrightarrow D)$.

Example 1.3.12: $p \Leftrightarrow q \models (p \vee (\boxed{p} \Rightarrow r)) \Leftrightarrow (p \vee (\boxed{q} \Rightarrow r))$.

Corollary 1.3.11

If $A \equiv B$
and by replacing A by B inside C we obtain D,
then $C \equiv D$.

(*i.e.* we can apply important equivalences anywhere inside formulas.)

Definitions

Definition 1.4.1

- A literal is a variable or its negation.
- A monomial is a conjunction of literals (special cases 0 and 1).
- A clause is a disjunction of literals (special cases 0 and 1).

Example 1.4.2

- $x, y, \neg z$ are literals.
- $x \wedge \neg y \wedge z$ is a monomial
- The monomial $x \wedge \neg y \wedge z \wedge \neg x$ contains x and $\neg x$: its value is 0.
- $x \vee \neg y \vee z$ is a clause
- The clause $x \vee \neg y \vee z \vee \neg x$ contains x and $\neg x$: its value is 1.

Normal form

Definition 1.4.3

A formula is in normal form if it only contains the operators $\wedge, \vee, \neg$ and the negations are only applied to variables.

Example 1.4.4

The formula $\neg a \vee b$ is in normal form
$a \Rightarrow b$ is not in normal form, even if it is equivalent to the first one.

Every formula admits an equivalent normal form.

Computing a normal form

1. **Equivalence elimination**
   Replace any occurrence of $A \Leftrightarrow B$ by
   (a) $(\neg A \vee B) \wedge (\neg B \vee A)$
   OR
   (b) $(A \wedge B) \vee (\neg A \wedge \neg B)$
2. **Implication elimination**
   Replace any occurrence of $A \Rightarrow B$ by $\neg A \vee B$
3. **Shifting negations towards variables**
   Replace any occurrence of
   (a) $\neg\neg A$ by A
   (b) $\neg(A \vee B)$ by $\neg A \wedge \neg B$
   (c) $\neg(A \wedge B)$ by $\neg A \vee \neg B$

How do we know this process is finite?

Proof by natural induction on the size of formulas.

Let us prove that any formula is equivalent to a formula without implication.
(The terminaison of the other steps can be proved similarly).

Base case

If $|A| = 0$, then A is $\top$ or $\perp$ or a variable.
By definition it does not contain any implication.

Remainder of the proof: inductive case

Induction hypothesis:

Assume the property holds for every formula with size $\leqslant n$,

and let us show it holds for every formula A with size $n+1$:

- Case 1: $A = \neg B$ with $|B| = n$.
  By induction hypothesis $B \equiv B'$ without implication.
  Hence $A = \neg B'$ which contains no implication.
- Case 2: $A = (B \circ C)$ with $|B| < n+1$ and $|C| < n+1$
  where $\circ$ is a binary connective.
  Again by induction hypothesis $B \equiv B'$ and $C \equiv C'$ without implications.
  We're down to two subcases:
  - if $\circ$ is an implication then $A \equiv \neg B' \vee C'$ without implication
  - if $\circ$ is another connective then $A \equiv B' \circ C'$ without implication

How to compute normal forms more efficiently

Simplify as soon as possible:

1. Replace $\neg(A \Rightarrow B)$ by $A \wedge \neg B$.
2. Replacing a conjunction by $\perp$ if it contains a formula and its negation
3. Replace a disjunction by $\top$ if it contains a formula and its negation
4. Apply :
    - Idempotence of $\wedge$ and $\vee$
    - Neutrality and absorption of $\perp$ and $\top$
    - Replace $\neg\top$ by $\perp$ and vice versa.
5. Apply the simplifications:
    - $x \vee (x \wedge y) \equiv x$,
    - $x \wedge (x \vee y) \equiv x$,
    - $x \vee (\neg x \wedge y) \equiv x \vee y$

Disjunctive normal form (DNF)

Definition 1.4.6

A formula is in disjunctive normal form (DNF) if and only if it is a disjunction (sum) of monomials.

Method: distribute the conjuctions over the disjuctions
$x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
The interest of a DNF is to highlight the models.

Example 1.4.7

$(x \wedge y) \vee (\neg x \wedge \neg y \wedge z)$ is a DNF, which has two main models:

- $x \mapsto 1, y \mapsto 1$
- $x \mapsto 0, y \mapsto 0, z \mapsto 1$

How to use DNFs for validity and countermodels

Transforming a formula into a disjunction of monomials also allows us to determine whether the formula is valid or not.

Let A be a formula whose validity we wish to check:

We transform $\neg A$ in an equivalent disjunction of monomials B:

- If $B = 0$ then $\neg A = 0$, hence $A = 1$, that is, A is valid
- Otherwise B is equal to a disjunction of nonzero monomials equivalent to $\neg A$, which give us models of $\neg A$, hence counter-models of A.

Example 1.4.9

Let $A = (p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \wedge q \Rightarrow r)$

Determine whether A is valid.

$\neg A$	
$\equiv (p \Rightarrow (q \Rightarrow r)) \wedge \neg(p \wedge q \Rightarrow r)$	since $\neg(B \Rightarrow C) \equiv B \wedge \neg C$
$\equiv (\neg p \vee \neg q \vee r) \wedge \neg(p \wedge q \Rightarrow r)$	eliminating two $\Rightarrow$
$\equiv (\neg p \vee \neg q \vee r) \wedge (p \wedge q \wedge \neg r)$	since $\neg(B \Rightarrow C) \equiv B \wedge \neg C$
$\equiv (\neg q \vee r) \wedge p \wedge q \wedge \neg r$	simplification $x \wedge (\neg x \vee y)$
$\equiv (r) \wedge p \wedge q \wedge \neg r$	simplification $x \wedge (\neg x \vee y)$
$= 0$	since we have $r \wedge \neg r$ in the monomial

Hence $\neg A = 0$ and $A = 1$, that is A is valid.

Example 1.4.10

Let $A = (a \Rightarrow b) \wedge c \vee (a \wedge d)$.

$$
\begin{array}{ll}
\neg A \equiv \neg((a \Rightarrow b) \wedge c) \wedge \neg(a \wedge d) & \text{(de Morgan)} \\
\equiv (\neg(a \Rightarrow b) \vee \neg c) \wedge (\neg a \vee \neg d) & \text{(de Morgan)} \\
\equiv ((a \wedge \neg b) \vee \neg c) \wedge (\neg a \vee \neg d) & \text{elimination of the implication} \\
\equiv (a \wedge \neg b \wedge \neg a) \vee (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d) & \\
 & \text{distributivity of } \vee \text{ over } \wedge \\
\equiv (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d) & \text{1st monomial contradictory}
\end{array}
$$

We obtain 3 models of $\neg A$: $(a \mapsto 1, b \mapsto 0, d \mapsto 0)$, $(a \mapsto 0, c \mapsto 0)$,
$(c \mapsto 0, d \mapsto 0)$.
That is, counter-models of A.
Hence A is not valid.

Conjunctive normal form (CNF)

Definition 1.4.11

A formula is a conjunctive normal form (CNF) if and only if it is a conjunction (product) of clauses.

Apply the (unusual) distributivity of disjunction over conjunction:

- $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$

The interest of a CNF is to highlight the counter-models.

Example 1.4.12

$(x \vee y) \wedge (\neg x \vee \neg y \vee z)$ is a CNF, which has two counter-models.

- $x \mapsto 0, y \mapsto 0$
- $x \mapsto 1, y \mapsto 1, z \mapsto 0.$

Utilisée également en modélisation (SAT-solvers)

Examples 1.4.8 and 1.4.13

Transformation in DNF of the following:

$$(a \vee b) \wedge (c \vee d \vee e) \equiv$$

$$\boxed{(a \wedge c) \vee (a \wedge d) \vee (a \wedge e) \vee (b \wedge c) \vee (b \wedge d) \vee (b \wedge e).}$$

Transformation in CNF of the following:

$$(a \wedge b) \vee (c \wedge d \wedge e) \equiv$$

$$\boxed{(a \vee c) \wedge (a \vee d) \wedge (a \vee e) \wedge (b \vee c) \wedge (b \vee d) \wedge (b \vee e).}$$

BDDC (*Binary Decision Diagram based Calculator*)

BDDC is a tool for manipulating propositional formulae developed by Pascal Raymond and available at the following address:

http://www-verimag.imag.fr/~raymond/home/tools/bddc/

Today

- Substitutions allow us to **deduce the validity** of a formula from another
- Replacements allow us to change part of a formula **without changing its meaning** and thus allow us to compute a simpler equivalent formula
- Every formula admits normal forms which allow to **highlight its models** and counter-models

Next course

- Boolean algebra
- Boolean functions
- Resolution

Prove our example by formula simplification

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