Logic formulae transformation

Stéphane Devismes  Pascal Lafourcade  Michel Lévy
Course given by Ioana Ciuciu (ciuciu@imag.fr)

Université Joseph Fourier, Grenoble I

18 January 2013
Conclusion : Today

- Introduction and Historical
- Propositional logic
- Syntax
- Truth value of formulae
- Important Equivalences
Our example with a truth table

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): Either Mary is the sister of John or Peter is old.

\[(p \Rightarrow \neg j) \land (\neg p \Rightarrow j) \land (j \Rightarrow m) \Rightarrow m \lor p\]
Our example with a truth table

**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): Either Mary is the sister of John or Peter is old.

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

<p>| | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>p</td>
<td>j</td>
<td>m</td>
<td>A = p ⇒ ¬j</td>
<td>B = ¬p ⇒ j</td>
<td>C = j ⇒ m</td>
<td>A∧B∧C</td>
<td>m∨p</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion

Logic formulae transformation

S. Devismes et al (Grenoble I)

18 January 2013
Preamble

How to prove that a formula is valid?

- Problem: for a formula having 100 variables, the truth table will contain $2^{100}$ lines (unable to be computed, even by a computer!).
- Idea: simplify the formula using substitutions, replacements, or normal form transformations (disjunctive or conjunctive).
- Then, solve the simplified formula using truth tables or a logic reasoning (for example: important equivalences).
Preamble

How to prove that a formula is valid?

- Truth table
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 (unable to be computed, even by a computer!).
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 (unable to be computed, even by a computer!).

- Idea:
  - Simplify the formula using substitutions, replacements, or normal form transformations (disjunctive or conjunctive)
  - Then, solve the simplified formula using truth tables or a logic reasoning (for example: important equivalences)
Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion
Substitution

Definition 1.3.1

A substitution $\sigma$ is a function mapping the set of variables into the set of formulae.
Substitution

Definition 1.3.1

A substitution $\sigma$ is a function mapping the set of variables into the set of formulae.

$A\sigma = \text{replacing all variables } x \text{ in the formula by the formula } \sigma(x)$. 
Substitution

**Definition 1.3.1**

A substitution $\sigma$ is a function mapping the set of variables into the set of formulae.

$A\sigma = $ replacing all variables $x$ in the formula by the formula $\sigma(x)$.

**Example:** $A = \neg(p \land q) \iff (\neg p \lor \neg q)$

- Let $\sigma$ the following substitution: $\sigma(p) = (a \lor b), \sigma(q) = (c \land d)$
- $A\sigma =$
Definition 1.3.1

A substitution $\sigma$ is a function mapping the set of variables into the set of formulae.

$A\sigma = $ replacing all variables $x$ in the formula by the formula $\sigma(x)$.

Example : $A = \neg(p \land q) \iff (\neg p \lor \neg q)$

- Let $\sigma$ the following substitution : $\sigma(p) = (a \lor b), \sigma(q) = (c \land d)$
- $A\sigma = \neg((a \lor b) \land (c \land d)) \iff (\neg(a \lor b) \lor \neg(c \land d))$
Finite support substitution

Definition 1.3.2 The support of a substitution $\sigma$

- The set of variables $x$ such as $x\sigma \neq x$.
- A substitution $\sigma$ which has finite support is denoted $< x_1 := A_1, \ldots, x_n := A_n >$, where $A_1, \ldots, A_n$ are formulae, $x_1, \ldots, x_n$ are distinct variables and the substitution verifies:
  - $\forall i, 1 \leq i \leq n : x_i\sigma = A_i$
  - $\forall y, y \not\in \{x_1, \ldots, x_n\} : y\sigma = y$

Example 1.3.3

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

$A\sigma = (a \lor b) \lor (a \lor b) \land y \Rightarrow (b \land c) \land y$
Finite support substitution

Definition 1.3.2 The support of a substitution $\sigma$

- The set of variables $x$ such as $x \sigma \neq x$.
- A substitution $\sigma$ which has finite support is denoted
  $< x_1 := A_1, \ldots, x_n := A_n >$, where $A_1, \ldots, A_n$ are formulae, $x_1, \ldots, x_n$ are distinct variables and the substitution verifies:
  
  $\forall i, i \in 1, \ldots, n : x_i \sigma = A_i$ \\
  $\forall y, y \not\in \{x_1, \ldots, x_n\} : y \sigma = y$

Example 1.3.3

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

$A \sigma =$
Logic formulae transformation
Substitution and replacement

Finite support substitution

**Definition 1.3.2 The support of a substitution \( \sigma \)**

- The set of variables \( x \) such as \( x \sigma \neq x \).
- A substitution \( \sigma \) which has **finite support** is denoted
  \(< x_1 := A_1, \ldots, x_n := A_n >\), where \( A_1, \ldots, A_n \) are formulae,
  \( x_1, \ldots, x_n \) are distinct variables and the substitution verifies:
  - \( \forall i, i \in 1, \ldots, n : x_i \sigma = A_i \)
  - \( \forall y, y \notin \{ x_1, \ldots, x_n \} : y \sigma = y \)

**Example 1.3.3**

\( A = x \lor x \land y \Rightarrow z \land y \) and \( \sigma = < x := a \lor b, z := b \land c > \)

\( A \sigma = (a \lor b) \lor (a \lor b) \land y \Rightarrow (b \land c) \land y \)
Properties of the substitutions

Property 1.3.4

Let \( A \) be a formula, \( \nu \) a truth assignment and \( \sigma \) a substitution, we have \( [A\sigma]_\nu = [A]_\nu \)
where for every variable \( x \), \( w(x) = [\sigma(x)]_\nu \).
Properties of the substitutions

Property 1.3.4

Let $A$ be a formula, $\nu$ a truth assignment and $\sigma$ a substitution, we have $[A\sigma]_{\nu} = [A]_{\nu}$ where for every variable $x$, $w(x) = [\sigma(x)]_{\nu}$.

**Example 1.3.5:**

Let $A = x \lor y \lor d$

Let $\sigma = < x \leftarrow a \lor b, y \leftarrow b \land c >$

Let $\nu$ so that $\nu(a) = 1$, $\nu(b) = 0$, $\nu(c) = 0$, $\nu(d) = 0$
Properties of the substitutions

Property 1.3.4

Let $A$ be a formula, $\nu$ a truth assignment and $\sigma$ a substitution, we have $[A\sigma]_\nu = [A]_w$ where for every variable $x$, $w(x) = [\sigma(x)]_\nu$.

Example 1.3.5 :
Let $A = x \lor y \lor d$
Let $\sigma = \langle x : = a \lor b, y : = b \land c \rangle$
Let $\nu$ so that $\nu(a) = 1, \nu(b) = 0, \nu(c) = 0, \nu(d) = 0$
$A\sigma = (a \lor b) \lor (b \land c) \lor d$
$[A\sigma]_\nu = (1 \lor 0) \lor (0 \land 0) \lor 0 = 1$
Properties of the substitutions

Property 1.3.4

Let $A$ be a formula, $\nu$ a truth assignment and $\sigma$ a substitution, we have $[A\sigma]_\nu = [A]_w$ where for every variable $x$, $w(x) = [\sigma(x)]_\nu$.

Example 1.3.5:
Let $A = x \lor y \lor d$
Let $\sigma = \langle x := a \lor b, y := b \land c \rangle$
Let $\nu$ so that $\nu(a) = 1$, $\nu(b) = 0$, $\nu(c) = 0$, $\nu(d) = 0$
$A\sigma = (a \lor b) \lor (b \land c) \lor d$
$[A\sigma]_\nu = (1 \lor 0) \lor (0 \land 0) \lor 0 = 1$
$w(x) = [\sigma(x)]_\nu = [a \lor b]_\nu = 1 \lor 0 = 1$
$w(y) = [\sigma(y)]_\nu = [b \land c]_\nu = 0 \land 0 = 0$
$w(d) = [\sigma(d)]_\nu = [d]_\nu = 0$
Properties of the substitutions

Property 1.3.4

Let $A$ be a formula, $\nu$ a truth assignment and $\sigma$ a substitution, we have $[A\sigma]_\nu = [A]_w$ where for every variable $x$, $w(x) = [\sigma(x)]_\nu$.

Example 1.3.5 :

Let $A = x \lor y \lor d$

Let $\sigma = \langle x := a \lor b, y := b \land c \rangle$

Let $\nu$ so that $\nu(a) = 1$, $\nu(b) = 0$, $\nu(c) = 0$, $\nu(d) = 0$

$A\sigma = (a \lor b) \lor (b \land c) \lor d$

$[A\sigma]_\nu = (1 \lor 0) \lor (0 \land 0) \lor 0 = 1$

$w(x) = [\sigma(x)]_\nu = [a \lor b]_\nu = 1 \lor 0 = 1$

$w(y) = [\sigma(y)]_\nu = [b \land c]_\nu = 0 \land 0 = 0$

$w(d) = [\sigma(d)]_\nu = [d]_\nu = 0$

$[A]_w = 1 \lor 0 \lor 0 = 1$
Properties of the substitutions

Property 1.3.4

Let $A$ be a formula, $\nu$ a truth assignment and $\sigma$ a substitution, we have $[A\sigma]_\nu = [A]_w$ where for every variable $x$, $w(x) = [\sigma(x)]_\nu$.

Example 1.3.5:

Let $A = x \lor y \lor d$
Let $\sigma = < x := a \lor b, y := b \land c >$
Let $\nu$ so that $\nu(a) = 1$, $\nu(b) = 0$, $\nu(c) = 0$, $\nu(d) = 0$

$A\sigma = (a \lor b) \lor (b \land c) \lor d$

$[A\sigma]_\nu = (1 \lor 0) \lor (0 \land 0) \lor 0 = 1$

$w(x) = [\sigma(x)]_\nu = [a \lor b]_\nu = 1 \lor 0 = 1$

$w(y) = [\sigma(y)]_\nu = [b \land c]_\nu = 0 \land 0 = 0$

$w(d) = [\sigma(d)]_\nu = [d]_\nu = 0$

$[A]_w = 1 \lor 0 \lor 0 = 1$

Démonstration.

Let $A$ a formula, $\nu$ a truth assignment and $\sigma$ a substitution.

Proof by induction on the height of the formulae.
Initial step: $|A| = 0$

Two possible cases:

- Let $A = k$ be a constant ($\top$ ou $\bot$): $[k\sigma]_v = [k]_v = [k]_w$. ($\top$ where $\bot$ equal 1 and 0 for all truth assignment)

- Let $A = x$ be a variable: by construction
  $[x\sigma]_v = [\sigma(x)]_v = w(x) = [x]_w$. 

Induction

Hypothesis: Suppose the property is true for all formula of height less or equal to \(n\).
Let \(A\) a formula of height \(n+1\); there are two possible cases:

- Case 1: Let \(A = \neg B\).
  \[
  [A_\sigma]_v = [\neg B_\sigma]_v = 1 - [B_\sigma]_v \quad \text{and} \quad [A]_w = [\neg B]_w = 1 - [B]_w.
  \]
  Since \(|B| = n\) we have \([B_\sigma]_v = [B]_w\)
  for all variable \(x, w(x) = [\sigma(x)]_v\).
  Hence, \([A_\sigma]_v = [A]_w\).
Induction

Hypothesis: Suppose the property is true for all formula of height less or equal to \( n \). Let \( A \) a formula of height \( n + 1 \); there are two possible cases:

- Case 2: Let \( A = B \circ C \), so then
  
  \[
  [A\sigma]_v = [B \circ C\sigma]_v = f_\circ([B\sigma]_v, [C\sigma]_v) \quad \text{and} \quad [A]_w = [B \circ C]_w = f_\circ([B]_w, [C]_w),
  \]
  where \( f_\circ \) is the function associated to \( \circ \) corresponding to definition 1.2.1.
  
  Since \(|B| < n + 1\) and \(|C| < n + 1\) we obtain by induction hypothesis \([B\sigma]_v = [B]_w\) and \([C\sigma]_v = [C]_w\) where for every variable \( x \), \( w(x) = [\sigma(x)]_v \), which implies that \([A\sigma]_v = [A]_w\).
Substitution of a valid formula

Theorem 1.3.6

The application of a substitution to a valid formula gives a valid formula.

Démonstration.

Let $A$ be a valid formula and $\sigma$ a substitution. Let $v$ be any truth assignment. According to property 1.3.4:

$[A_{\sigma}]_v = [A]_w$ where for every variable $x$,

$w(x) = [\sigma(x)]_v$.

Since $A$ is valid, $[A]_w = 1$. Consequently, $A_{\sigma}$ equals 1 in every truth assignment, it is therefore a valid formula.
Substitution of a valid formula

**Theorem 1.3.6**

The application of a substitution to a valid formula gives a valid formula.

**Déémonstration.**

Let $A$ be a valid formula and $\sigma$ a substitution. Let $\nu$ be any truth assignment.

According to property 1.3.4:

\[ A_{\sigma}^{\nu} = A_{w}^{\nu} \]

where for every variable $x$, $w(x) = \sigma(x)^{\nu}$. Since $A$ is valid, $A_{w}^{\nu} = 1$. Consequently, $A_{\sigma}^{\nu}$ equals 1 in every truth assignment, it is therefore a valid formula.
Substitution of a valid formula

Theorem 1.3.6
The application of a substitution to a valid formula gives a valid formula.

Démonstration.
Let $A$ be a valid formula and $\sigma$ a substitution.
Let $\nu$ be any truth assignment.

According to property 1.3.4: $[A\sigma]_\nu = [A]_w$ where for every variable $x$, $w(x) = [\sigma(x)]_\nu$. 
Substitution of a valid formula

Theorem 1.3.6

The application of a substitution to a valid formula gives a valid formula.

Démonstration.

Let $A$ be a valid formula and $\sigma$ a substitution. Let $\nu$ be any truth assignment.

According to property 1.3.4: $[A\sigma]_{\nu} = [A]_{w}$ where for every variable $x$, $w(x) = [\sigma(x)]_{\nu}$.

Since $A$ is valid, $[A]_{w} = 1$. Consequently, $A\sigma$ equals 1 in every truth assignment, it is therefore a valid formula.
Examples

Example

Using substitution, prove that $F = (a \land b) \lor \neg a \lor \neg b$ is valid.

Let $A$ the formula $x \lor \neg x$ is valid

Let $\sigma$ the following substitution : $\sigma(x) = (a \land b)$

The formula $A_{\sigma}$ is $(a \land b) \lor \neg (a \land b)$

According to theorem 1.3.6, $A_{\sigma}$ is valid

From the De Morgan laws, we have $A_{\sigma} = (a \land b) \lor \neg a \lor \neg b$"
Examples

Example

Using substitution, prove that $F = (a \land b) \lor \neg a \lor \neg b$ is valid.

- Let $A$ the formula $x \lor \neg x$
- $A$ is valid
Examples

Example

Using substitution, prove that $F = (a \land b) \lor \neg a \lor \neg b$ is valid.

- Let $A$ the formula $x \lor \neg x$
- $A$ is valid
- Let $\sigma$ the following substitution: $\sigma(x) = (a \land b)$
- The formula $A\sigma$ is $(a \land b) \lor \neg (a \land b)$
Examples

Example

Using substitution, prove that $F = (a \land b) \lor \neg a \lor \neg b$ is valid.

- Let $A$ the formula $x \lor \neg x$
- $A$ is valid
- Let $\sigma$ the following substitution : $\sigma(x) = (a \land b)$
- The formula $A\sigma$ is $(a \land b) \lor \neg (a \land b)$
- According to theorem 1.3.6, $A\sigma$ is valid
Examples

Example

Using substitution, prove that $F = (a \land b) \lor \neg a \lor \neg b$ is valid.

- Let $A$ the formula $x \lor \neg x$
- $A$ is valid
- Let $\sigma$ the following substitution: $\sigma(x) = (a \land b)$
- The formula $A\sigma$ is $(a \land b) \lor \neg(a \land b)$
- According to theorem 1.3.6, $A\sigma$ is valid
- From the De Morgan laws, we have $A\sigma = (a \land b) \lor (\neg a \lor \neg b)$
- Hence, $A\sigma = (a \land b) \lor \neg a \lor \neg b$ is valid
Examples

Example 1.3.7

Let $A$ the formula $\neg(p \land q) \iff (\neg p \lor \neg q)$. This formula is valid, it is an important equivalence. Let $\sigma$ the following substitution : $< p := (a \lor b), q := (c \land d) >$. The formula $A\sigma =$
Examples

Example 1.3.7

Let $A$ the formula $\neg(p \land q) \iff (\neg p \lor \neg q)$. This formula is valid, it is an important equivalence. Let $\sigma$ the following substitution :

\[
\begin{align*}
& < p := (a \lor b), q := (c \land d) >. \\
\text{The formula } A\sigma = \\
& \neg((a \lor b) \land (c \land d)) \iff (\neg(a \lor b) \lor \neg(c \land d)) \text{ is also valid.}
\end{align*}
\]
Replacement

Replace a formula by another formula.

Definition 1.3.8

Let $A, B, C, D$ formulae.
The formula $D$ is obtained by replacing in $C$ certain occurrences of $A$ by $B$
if there exist a formula $E$ and a variable $x$ so that, $C = E < x := A >$
and $D = E < x := B >$. 
Examples

Example 1.3.9

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

- The formula obtained by replacing all occurrences of $(a \Rightarrow b)$ by $(a \land b)$ in $C$ is
Examples

Example 1.3.9

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

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

  $D = ((a \land b) \lor \neg(a \land b))$,

  it is obtained considering the formula $E = (x \lor \neg x)$ and the following substitutions $< x := (a \land b) >$ et $< x := (a \Rightarrow b) >$. 
Example 1.3.9

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

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

$$D = ((a \land b) \lor \neg(a \land b)),$$

it is obtained considering the formula $E = (x \lor \neg x)$ and the following substitutions $< x := (a \land b) >$ et $< x := (a \Rightarrow b) >$.

- The formula obtained by replacing the first occurrence of $(a \Rightarrow b)$ by $(a \land b)$ in $C$ is
Examples

Example 1.3.9

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

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

  $$D = ((a \land b) \lor \neg(a \land b)),$$

  it is obtained considering the formula $E = (x \lor \neg x)$ and the following substitutions $< x := (a \land b) >$ et $< x := (a \Rightarrow b) >$.

- The formula obtained by replacing the first occurrence of $(a \Rightarrow b)$ by $(a \land b)$ in $C$ is

  $$D = ((a \land b) \lor \neg(a \Rightarrow b)),$$

  it is obtained considering the formula $E = (x \lor \neg(a \Rightarrow b))$ and the following substitution $< x := (a \land b) >$ and $< x := (a \Rightarrow b) >$. 
Properties of the replacements (1/2)

**Theorem 1.3.10**

Let $C$ a formula and $D$ the formula obtained by replacing, in $C$, the occurrences of formula $A$ by formula $B$. We have:

$$(A \iff B) \models (C \iff D).$$
Theorem 1.3.10

Let $C$ a formula and $D$ the formula obtained by replacing, in $C$, the occurrences of formula $A$ by formula $B$. We have:

$$(A \Leftrightarrow B) \models (C \Leftrightarrow D).$$

Démonstration.

By definition of the replacement, there is a formula $E$ and a variable $x$ so that, $C = E < x := A >$ et $D = E < x := B >$. Suppose that $\nu$ is a model truth assignment of $(A \Leftrightarrow B)$. We therefore have $[A]_\nu = [B]_\nu$.

According to property 1.3.4:

1. $[C]_\nu = [E]_w$ where $w$ is identical to $\nu$ except that $w(x) = [A]_\nu$
2. $[D]_\nu = [E]_{w'}$ where $w'$ is identical to $\nu$ except that $w'(x) = [B]_\nu$

Since $[A]_\nu = [B]_\nu$, the truth assignments $w$ and $w'$ are identical, therefore $[C]_\nu = [D]_\nu$. Consequently, $\nu$ is a model of $(C \Leftrightarrow D)$. \qed
Application of the theorem (Example 1.3.12)

\[ p \iff q \models (p \lor (\overline{p} \Rightarrow r)) \iff (p \lor (q \Rightarrow r)). \]
Properties of the replacements (2/2)

Corollary 1.3.11

Let $C$ a formula and $D$ the formula obtained by replacing, in $C$, one occurrence of formula $A$ by formula $B$. We have : if $A \equiv B$ then $C \equiv D$. 

Example 1.3.12

$(\neg(p \lor q) \Rightarrow (\neg(p \lor q) \lor r)) \equiv (\neg(p \lor q) \Rightarrow (\neg p \land \neg q) \lor r)$, since $\neg(p \lor q) \equiv (\neg p \land \neg q)$. 

S. Devismes et al (Grenoble I)
Properties of the replacements (2/2)

Corollary 1.3.11

Let $C$ a formula and $D$ the formula obtained by replacing, in $C$, one occurrence of formula $A$ by formula $B$. We have : if $A \equiv B$ then $C \equiv D$.

Démonstration.

If $A \equiv B$, then the formula $(A \leftrightarrow B)$ is valid (property 1.2.10), hence the formula $(C \leftrightarrow D)$ is also valid since, according to theorem 1.3.10, the consequence of $(A \leftrightarrow B)$. Consequently $C \equiv D$. \qed
Corollary 1.3.11

Let $C$ a formula and $D$ the formula obtained by replacing, in $C$, one occurrence of formula $A$ by formula $B$. We have: if $A \equiv B$ then $C \equiv D$.

Démonstration.

If $A \equiv B$, then the formula $(A \iff B)$ is valid (property 1.2.10), hence the formula $(C \iff D)$ is also valid since, according to theorem 1.3.10, the consequence of $(A \iff B)$. Consequently $C \equiv D$.

Example 1.3.12

$\neg(p \lor q) \Rightarrow (\neg(p \lor q) \lor r) \equiv (\neg(p \lor q) \Rightarrow (\neg p \land \neg q) \lor r)$, since $\neg(p \lor q) \equiv (\neg p \land \neg q)$. 
Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion
Definitions

Definition 1.4.1

- A literal is a variable or its negation.
Definitions

Definition 1.4.1

- A literal is a variable or its negation.
- A monome is a conjunction of literals.
Definitions

Definition 1.4.1

- A literal is a variable or its negation.
- A monome is a conjunction of literals.
- A clause is a disjunction of literals.
Example 1.4.2

- \( x, y, \neg z \) are literals.
Example 1.4.2

- $x, y, \neg z$ are literals.
- $x \land \neg y \land z$ is a monome whose unique model is $x = 1, y = 0, z = 1$. 
Example 1.4.2

- $x, y, \neg z$ are literals.
- $x \land \neg y \land z$ is a monome whose unique model is $x = 1, y = 0, z = 1$.
- The monome $x \land \neg y \land z \land \neg x$ contains a variable and its negation: it equals 0.
Example 1.4.2

- $x, y, \neg z$ are literals.
- $x \land \neg y \land z$ is a monome whose unique model is $x = 1, y = 0, z = 1$.
- The monome $x \land \neg y \land z \land \neg x$ contains a variable and its negation: it equals 0.
- $x \lor \neg y \lor z$ is a clause whose unique counter model is $x = 0, y = 1, z = 0$. 
Example 1.4.2

- $x, y, \neg z$ are literals.
- $x \land \neg y \land z$ is a monome whose unique model is $x = 1, y = 0, z = 1$.
- The monome $x \land \neg y \land z \land \neg x$ contains a variable and its negation: it equals 0.
- $x \lor \neg y \lor z$ is a clause whose unique counter model is $x = 0, y = 1, z = 0$.
- The clause $x \lor \neg y \lor z \lor \neg x$ contains a variable and its negation: it equals 1.
Normal form

Definition 1.4.3

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

Definition 1.4.3
A formula is in normal form if it only contains the operators $\land$, $\lor$, $\neg$ and the negations are only applied to variables.

Example 1.4.4
The formula $\neg a \lor b$ is in normal form, while the formula $a \Rightarrow b$ is not in normal form, even if it is equivalent to the first one.
Normal form

Definition 1.4.3

A formula is in normal form if it only contains the operators $\land$, $\lor$, $\lnot$ and the negations are only applied to variables.

Example 1.4.4

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

1. Equivalence elimination
2. Implication elimination
3. Shifting negations such that they only apply to variables
1. Eliminating an equivalence

Replacing an occurrence of $A \iff B$ by one of the sub-formulae

(a) $(\neg A \lor B) \land (\neg B \lor A)$

(b) $(A \land B) \lor (\neg A \land \neg B)$
Eliminating an implication

Replacing an occurrence of $A \Rightarrow B$ by $\neg A \lor B$
Shifting negations

Replacing an occurrence of

(a) \( \neg \neg A \) by \( A \)

(b) \( \neg (A \lor B) \) by \( \neg A \land \neg B \)

(c) \( \neg (A \land B) \) by \( \neg A \lor \neg B \)
Remark 1.4.5 : simplifications

Simplify as soon as possible :

1. Replace a sub-formula of the form $\neg(A \Rightarrow B)$ by $A \land \neg B$. 

2. Replacing a conjunction by 0 if it contains
   - either a formula and its negation,
   - or a 0

3. Replace a disjunction by 1, if it contains
   - either a formula and its negation,
   - or a 1

4. Replace $\neg1$ by 0 and $\neg0$ by 1

5. Eliminate the 0 from the disjunctions and the 1 from the conjunctions

6. Apply the simplifications :
   - $x \lor (x \land y) = x$
   - $x \land (x \lor y) = x$
   - $x \lor (\neg x \land y) = x \lor y$

7. Apply the idempotence of the conjunction and the disjunction.
Remark 1.4.5: simplifications

Simplify as soon as possible:

1. Replace a sub-formula of the form \( \neg (A \Rightarrow B) \) by \( A \land \neg B \).

2. Replacing a conjunction by 0 if it contains
   ▶ either a formula and its negation,
   ▶ or a 0

3. Replace a disjunction by 1, if it contains
   ▶ either a formula and its negation,
   ▶ or a 1

4. Replace \( \neg 1 \) by 0 and \( \neg 0 \) by 1

5. Eliminate the 0 from the disjunctions and the 1 from the conjunctions

6. Apply the simplifications:
   ▶ \( x \lor (x \land y) = x \),
   ▶ \( x \land (x \lor y) = x \),
   ▶ \( x \lor (\neg x \land y) = x \lor y \)
Remark 1.4.5 : simplifications

Simplify as soon as possible :

1. Replace a sub-formula of the form \( \neg(A \Rightarrow B) \) by \( A \land \neg B \).

2. Replacing a conjunction by 0 if it contains
   - either a formula and its negation,
   - or a 0

3. Replace a disjunction by 1, if it contains
   - either a formula and its negation,
   - or a 1
Remark 1.4.5 : simplifications

Simplify as soon as possible :

1. Replace a sub-formula of the form $\neg(A \Rightarrow B)$ by $A \land \neg B$.

2. Replacing a conjunction by 0 if it contains
   - either a formula and its negation,
   - or a 0

3. Replace a disjunction by 1, if it contains
   - either a formula and its negation,
   - or a 1

4. Replace $\neg 1$ by 0 and $\neg 0$ by 1
Remark 1.4.5: simplifications

Simplify as soon as possible:

1. Replace a sub-formula of the form \( \neg (A \Rightarrow B) \) by \( A \land \neg B \).

2. Replacing a conjunction by 0 if it contains
   - either a formula and its negation,
   - or a 0

3. Replace a disjunction by 1, if it contains
   - either a formula and its negation,
   - or a 1

4. Replace \( \neg 1 \) by 0 and \( \neg 0 \) by 1

5. Eliminate the 0 from the disjunctions and the 1 from the conjunctions
Remark 1.4.5 : simplifications

Simplify as soon as possible :

1. Replace a sub-formula of the form \( \neg (A \Rightarrow B) \) by \( A \land \neg B \).
2. Replacing a conjunction by 0 if it contains
   - either a formula and its negation,
   - or a 0
3. Replace a disjunction by 1, if it contains
   - either a formula and its negation,
   - or a 1
4. Replace \( \neg 1 \) by 0 and \( \neg 0 \) by 1
5. Eliminate the 0 from the disjunctions and the 1 from the conjunctions
6. Apply the simplifications :
   - \( x \lor (x \land y) = x \),
   - \( x \land (x \lor y) = x \),
   - \( x \lor (\neg x \land y) = x \lor y \)
Remark 1.4.5 : simplifications

Simplify as soon as possible :

1. Replace a sub-formula of the form \( \neg (A \Rightarrow B) \) by \( A \land \neg B \).

2. Replacing a conjunction by 0 if it contains
   - either a formula and its negation,
   - or a 0

3. Replace a disjunction by 1, if it contains
   - either a formula and its negation,
   - or a 1

4. Replace \( \neg 1 \) by 0 and \( \neg 0 \) by 1

5. Eliminate the 0 from the disjunctions and the 1 from the conjunctions

6. Apply the simplifications :
   - \( x \lor (x \land y) = x \),
   - \( x \land (x \lor y) = x \),
   - \( x \lor (\neg x \land y) = x \lor y \)

7. Apply the idempotence of the conjunction and the disjunction.
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 monomes.

Distribution of conjuctions over disjunctions

\[ x \land (y \lor z) = (x \land y) \lor (x \land z) \]
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 monomes.

Distribution of conjunctions over disjunctions

\[ x \land (y \lor z) = (x \land y) \lor (x \land z) \]

The interest of DNFs is to highlight their models.
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 monomes.

Distribution of conjunctions over disjunctions

\[ x \land (y \lor z) = (x \land y) \lor (x \land z) \]

The interest of DNFs is to highlight their models.

**Example 1.4.7**

\((x \land y) \lor (\neg x \land \neg y \land z)\) is a DNF, which has two models
**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 monomes.

**Distribution of conjunctions over disjunctions**

\[ x \land (y \lor z) = (x \land y) \lor (x \land z) \]

The interest of DNFs is to highlight their models.

**Example 1.4.7**

\((x \land y) \lor (\neg x \land \neg y \land z)\) is a DNF, which has two models

- \(x = 1, y = 1\)
- \(x = 0, y = 0, z = 1\)
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.

Applying distributivity (unusual) of disjunction over conjunction:

- \( A \lor (B \land C) = (A \lor B) \land (A \lor C) \)
- \( (B \land C) \lor A = (B \lor A) \land (C \lor A) \).
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.

Applying distributivity (unusual) of disjunction over conjunction:

- $A \lor (B \land C) = (A \lor B) \land (A \lor C)$
- $(B \land C) \lor A = (B \lor A) \land (C \lor A)$.

The interest of CNF is to highlight their counter-models.
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.

Applying distributivity (unusual) of disjunction over conjunction:

- $A \lor (B \land C) = (A \lor B) \land (A \lor C)$
- $(B \land C) \lor A = (B \lor A) \land (C \lor A)$.

The interest of CNF is to highlight their counter-models.

Example 1.4.12

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

Applying distributivity (unusual) of disjunction over conjunction:

\[
A \lor (B \land C) = (A \lor B) \land (A \lor C)
\]

\[
(B \land C) \lor A = (B \lor A) \land (C \lor A).
\]

The interest of CNF is to highlight their counter-models.

**Example 1.4.12**

\[(x \lor y) \land (\neg x \lor \neg y \lor z)\] is a CNF, which has two counter-models.

- \(x = 0, y = 0\)
- \(x = 1, y = 1, z = 0.\)
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

\[
\neg a \lor b \iff \neg \neg b \lor \neg a \iff \neg a \lor b \land b \lor \neg a \iff \neg a \land b \lor \neg a \lor b \land \neg a \lor b \iff \neg a \lor b \lor \neg b \lor \neg a \iff 1 \lor b \iff 1.
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (\(DNF\)):

\[
\neg a \lor b \iff \neg \neg b \lor \neg a
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

\[
\begin{align*}
\text{▶} & \quad (\neg a \vee b) \iff (\neg \neg b \vee \neg a) \\
\text{▶} & \quad (\neg a \vee b) \iff (b \vee \neg a)
\end{align*}
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

- \((-a \lor b) \iff (\neg \neg b \lor \neg a)\)
- \((-a \lor b) \iff (b \lor \neg a)\)
- \((-a \lor b) \land (b \lor \neg a) \lor \neg(-a \lor b) \land \neg(b \lor \neg a)\)
Example

Transform the formula $(a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)$ in disjunction of monomes (DNF):

- $(\neg a \lor b) \iff (\neg \neg b \lor \neg a)$
- $(\neg a \lor b) \iff (b \lor \neg a)$
- $(\neg a \lor b) \land (b \lor \neg a) \lor \neg(\neg a \lor b) \land \neg(b \lor \neg a)$
- $(\neg a \lor b) \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a)$
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

\[
\begin{align*}
\neg a \lor b & \iff \neg \neg b \lor \neg a \\
\neg a \lor b & \iff b \lor \neg a \\
\neg a \lor b \land (b \lor \neg a) \lor \neg (\neg a \lor b) \land \neg (b \lor \neg a) & \\
\neg a \lor b \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a) & \\
\neg a \land b \lor (\neg a \land \neg a) \lor (b \land b) \lor (b \land \neg a) \lor (a \land \neg b \land \neg b \land a) & 
\end{align*}
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

\[
\begin{align*}
\triangleright &\quad (\neg a \lor b) \iff (\neg \neg b \lor \neg a) \\
\triangleright &\quad (\neg a \lor b) \iff (b \lor \neg a) \\
\triangleright &\quad (\neg a \lor b) \land (b \lor \neg a) \lor \neg (\neg a \lor b) \land \neg (b \lor \neg a) \\
\triangleright &\quad (\neg a \lor b) \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a) \\
\triangleright &\quad (\neg a \land b) \lor (\neg a \land \neg a) \lor (b \land b) \lor (b \land \neg a) \lor (a \land \neg b \land \neg b \land a) \\
\triangleright &\quad (\neg a \land b) \lor \neg a \lor b \lor (b \land \neg a) \lor (a \land \neg b) 
\end{align*}
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

1. \((-a \lor b) \iff (-\neg b \lor \neg a)\)
2. \((-a \lor b) \iff (b \lor \neg a)\)
3. \((-a \lor b) \land (b \lor \neg a) \lor \neg(-a \lor b) \land \neg(b \lor \neg a)\)
4. \((-a \lor b) \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a)\)
5. \((-a \land b) \lor (-a \land \neg a) \lor (b \land b) \lor (b \land \neg a) \lor (a \land \neg b \land \neg b \land a)\)
6. \((-a \land b) \lor \neg a \lor b \lor (b \land \neg a) \lor (a \land \neg b)\)
7. \(-a \lor b \lor (a \land \neg b), \text{ since } x \lor (x \land y) = x\)
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

\[
\begin{align*}
&(\neg a \lor b) \iff (\neg \neg b \lor \neg a) \\
&(\neg a \lor b) \iff (b \lor \neg a) \\
&(\neg a \lor b) \land (b \lor \neg a) \lor \neg (\neg a \lor b) \land \neg (b \lor \neg a) \\
&(\neg a \lor b) \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a) \\
&(\neg a \land b) \lor (\neg a \land \neg a) \lor (b \land b) \lor (b \land \neg a) \lor (a \land \neg b \land \neg b \land a) \\
&(\neg a \land b) \lor \neg a \lor b \lor (b \land \neg a) \lor (a \land \neg b) \\
&\neg a \lor b \lor (a \land \neg b), \text{ since } x \lor (x \land y) = x \\
&\neg a \lor b \lor a, \text{ since } x \lor (\neg x \land y) = x \lor y
\end{align*}
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (DNF):

\[
\begin{align*}
&\therefore (\neg a \lor b) \iff (\neg \neg b \lor \neg a) \\
&\therefore (\neg a \lor b) \iff (b \lor \neg a) \\
&\therefore (\neg a \lor b) \land (b \lor \neg a) \lor \neg (\neg a \lor b) \land \neg (b \lor \neg a) \\
&\therefore (\neg a \lor b) \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a) \\
&\therefore (\neg a \land b) \lor (\neg a \land \neg a) \lor (b \land b) \lor (b \land \neg a) \lor (a \land \neg b \land \neg b \land a) \\
&\therefore (\neg a \land b) \lor \neg a \lor b \lor (b \land \neg a) \lor (a \land \neg b) \\
&\therefore \neg a \lor b \lor (a \land \neg b), \text{ since } x \lor (x \land y) = x \\
&\therefore \neg a \lor b \lor a, \text{ since } x \lor (\neg x \land y) = x \lor y \\
&\therefore 1 \lor b
\end{align*}
\]
Example

Transform the formula \((a \Rightarrow b) \iff (\neg b \Rightarrow \neg a)\) in disjunction of monomes (\(DNF\)):

\[
\begin{align*}
\quad & (\neg a \lor b) \iff (\neg \neg b \lor \neg a) \\
\quad & (\neg a \lor b) \iff (b \lor \neg a) \\
\quad & (\neg a \lor b) \land (b \lor \neg a) \lor \neg (\neg a \lor b) \land \neg (b \lor \neg a) \\
\quad & (\neg a \lor b) \land (b \lor \neg a) \lor (a \land \neg b) \land (\neg b \land a) \\
\quad & (\neg a \land b) \lor (\neg a \land \neg a) \lor (b \land b) \lor (b \land \neg a) \lor (a \land \neg b \land \neg b \land a) \\
\quad & (\neg a \land b) \lor \neg a \lor b \lor (b \land \neg a) \lor (a \land \neg b) \\
\quad & \neg a \lor b \lor (a \land \neg b), \text{ since } x \lor (x \land y) = x \\
\quad & \neg a \lor b \lor a, \text{ since } x \lor (\neg x \land y) = x \lor y \\
\quad & 1 \lor b \\
\quad & 1
\end{align*}
\]
Example 1.4.8 et 1.4.13

Transformation in DNF of the following:

\[(a \lor b) \land (c \lor d \lor e) \equiv\]
Example 1.4.8 et 1.4.13

Transformation in DNF of the following:

\[(a \lor b) \land (c \lor d \lor e) \equiv\]
\[(a \land c) \lor (a \land d) \lor (a \land e) \lor (b \land c) \lor (b \land d) \lor (b \land e).\]
Example 1.4.8 et 1.4.13

Transformation in **DNF** of the following:

\[(a \lor b) \land (c \lor d \lor e) \equiv (a \land c) \lor (a \land d) \lor (a \land e) \lor (b \land c) \lor (b \land d) \lor (b \land e).\]

Transformation in **CNF** of the following:

\[(a \land b) \lor (c \land d \land e) \equiv\]
Example 1.4.8 et 1.4.13

Transformation in **DNF** of the following :

\[(a \lor b) \land (c \lor d \lor e) \equiv (a \land c) \lor (a \land d) \lor (a \land e) \lor (b \land c) \lor (b \land d) \lor (b \land e).\]

Transformation in **CNF** of the following :

\[(a \land b) \lor (c \land d \land e) \equiv (a \lor c) \land (a \lor d) \land (a \lor e) \land (b \lor c) \land (b \lor d) \land (b \lor e).\]
Utilisation of disjunctions of monomes

PURPOSE is to transform a formula in DNF

Determine if a formula is valid or not.
Utilisation of disjunctions of monomes

**PURPOSE** is to transform a formula in DNF

Determine if a formula is valid or not.

Let $A$ a formula whose validity we wish to check:
Utilisation of disjunctions of monomes

**PURPOSE** is to transform a formula in DNF

Determine if a formula is valid or not.

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

We transform $\neg A$ in an *equivalent* disjunction of monomes $B$.
Utilisation of disjunctions of monomes

**PURPOSE** is to transform a formula in DNF

Determine if a formula is valid or not.

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

We transform $\lnot A$ in an *equivalent* disjunction of monomes $B$

- If $B = 0$ then $\lnot A = 0$, hence $A = 1$, that is, $A$ is valid
Utilisation of disjunctions of monomes

PURPOSE is to transform a formula in DNF

Determine if a formula is valid or not.

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

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

- If $B = 0$ then $\neg A = 0$, hence $A = 1$, that is, $A$ is valid
- Else $B$ is equal to a disjunction of nonzero monomes equivalent to $\neg A$, which gives us models of $\neg A$, hence counter-models of $A$. 
Example 1.4.9

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

Determine if $A$ is valid.
Example 1.4.9

Let \( A = (p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \land q \Rightarrow r) \)

Determine if \( A \) is valid.

\[
\neg A = (p \Rightarrow (q \Rightarrow r)) \land \neg (p \land q \Rightarrow r)
\]

\[
= (\neg p \lor \neg q \lor r) \land \neg (p \land q \Rightarrow r)
\]

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

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

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

\[
= 0
\]

Hence \( \neg A = 0 \) and \( A = 1 \), that is \( A \) is valid.
Example 1.4.10

Let $A = (a \Rightarrow b) \land c \lor (a \land d)$.

Determine if $A$ is valid.
Example 1.4.10

Let \( A = (a \Rightarrow b) \land c \lor (a \land d) \).

Determine if \( A \) is valid.

\[
\neg A \\
= \neg((a \Rightarrow b) \land c) \land \neg(a \land d) \\
= (\neg(a \Rightarrow b) \lor \neg c) \land (\neg a \lor \neg d) \\
= ((a \land \neg b) \lor \neg c) \land (\neg a \lor \neg d) \\
= (a \land \neg b \land \neg a) \lor (a \land \neg b \land \neg d) \
\lor (\neg c \land \neg a) \lor (\neg c \land \neg d) \\
= (a \land \neg b \land \neg d) \lor (\neg c \land \neg a) \lor (\neg c \land \neg d)
\]

shifting the negations
shifting the negations
shifting of one negation
elimination of the implication
distributivity of the disjunction
over the conjunction
simplification

We obtain 3 models of \( \neg A \): \( (a = 1, b = 0, d = 0) \), \( (a = 0, c = 0) \), \( (c = 0, d = 0) \).

That is, counter-models of \( A \).

Hence \( A \) is not valid.
Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion
Définition 1.5.1

A Boolean Algebra is a set of at least two elements, 0, 1, and three operations, complement (negation) ($\overline{x}$), sum (disjunction) (+) and product (conjunction) ($\cdot$), which verify the following axioms:

1. the sum is:
   - associative: $x + (y + z) = (x + y) + z$,
   - commutative: $x + y = y + x$,
   - 0 is the neutral element for sum: $0 + x = x$,
Définition 1.5.1

A Boolean Algebra is a set of at least two elements, 0, 1, and three operations, complement (negation) \( \overline{x} \), sum (disjunction) \( + \) and product (conjunction) \( . \), which verify the following axioms:

1. the sum is:
   - associative: \( x + (y + z) = (x + y) + z \),
   - commutative: \( x + y = y + x \),
   - 0 is the neutral element for sum: \( 0 + x = x \),

2. the product is:
   - associative: \( x.(y.z) = (x.y).z \),
   - commutative: \( x.y = y.x \),
   - 1 is the neutral element for product: \( 1.x = x \),
Définition 1.5.1

A Boolean Algebra is a set of at least two elements, 0, 1, and three operations, complement (negation) \((\overline{x})\), sum (disjunction) \((+\)) and product (conjunction) \((.)\), which verify the following axioms:

1. the sum is:
   - associative: \(x + (y + z) = (x + y) + z\),
   - commutative: \(x + y = y + x\),
   - 0 is the neutral element for sum: \(0 + x = x\),

2. the product is:
   - associative: \(x.(y.z) = (x.y).z\),
   - commutative: \(x.y = y.x\),
   - 1 is the neutral element for product: \(1.x = x\),

3. the product is distributive over the sum: \(x.(y + z) = (x.y) + (x.z)\),

4. the sum is distributive over the product: \(x + (y.z) = (x + y).(x + z)\),
Définition 1.5.1

A Boolean Algebra is a set of at least two elements, 0, 1, and three operations, complement (negation) (\(\overline{x}\)), sum (disjunction) (\(+\)) and product (conjunction) (\(\cdot\)), which verify the following axioms:

1. the sum is:
   - associative: \(x + (y + z) = (x + y) + z\),
   - commutative: \(x + y = y + x\),
   - 0 is the neutral element for sum: \(0 + x = x\),

2. the product is:
   - associative: \(x \cdot (y \cdot z) = (x \cdot y) \cdot z\),
   - commutative: \(x \cdot y = y \cdot x\),
   - 1 is the neutral element for product: \(1 \cdot x = x\),

3. the product is distributive over the sum:
   \(x \cdot (y + z) = (x \cdot y) + (x \cdot z)\),

4. the sum is distributive over the product:
   \(x + (y \cdot z) = (x + y) \cdot (x + z)\),

5. negation laws:
   - \(x + \overline{x} = 1\),
   - \(x \cdot \overline{x} = 0\).
Propositional logic is a Boolean Algebra

The axioms can be proven by truth tables.
Propositional logic is a Boolean Algebra

The axioms can be proven by truth tables.

Another example:

<table>
<thead>
<tr>
<th>Boolean Algebra</th>
<th>$P(X)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$X$</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>$\overline{p}$</td>
<td>$X - p$</td>
</tr>
<tr>
<td>$p + q$</td>
<td>$p \cup q$</td>
</tr>
<tr>
<td>$p.q$</td>
<td>$p \cap q$</td>
</tr>
</tbody>
</table>

**Figure** : Figure 1.1
Property of a Boolean Algebra

Property 1.5.3

- For all $x$, there is one and only one $y$ so that $x + y = 1$ and $xy = 0$, in other words, the negation is unique.
(proof can be found in the course support (poly))
Property of a Boolean Algebra

Property 1.5.3

- For all \( x \), there is one and only one \( y \) so that \( x + y = 1 \) and \( xy = 0 \), in other words, the negation is unique.
  (proof can be found in the course support (poly))

- \( \bar{1} = 0 \)
- \( \bar{0} = 1 \)
- \( \bar{x} = x \)
- Product idempotence: \( x.x = x \)
- Sum idempotence: \( x + x = x \)
- 1 is absorbing element for sum: \( 1 + x = 1 \)
- 0 is absorbing element for product: \( 0.x = 0 \)
- De Morgan laws:
  - \( \bar{xy} = \bar{x} + \bar{y} \)
  - \( \bar{x + y} = \bar{x} \bar{y} \)
Proof

1. $\overline{1} = 0$. 

According to the definition of the negation, $x \cdot x = 0$. Hence, $1 \cdot 1 = 0$. Since 1 is neutral element for product, we have $1 = 0$. 

2. $0 = 1$. 

According to the definition of the negation, $x + x = 1$. Hence, $0 + 0 = 1$. Since 0 is neutral element for sum, we have $0 = 1$. 

3. $\overline{x} = x$. 

According to the properties of negation and commutativity, we have: $x + x = 1$, $x \cdot x = 0$, $x + x = 1$, and $x \cdot x = 0$. Because of the uniqueness of the negation (property 1.5.3), we deduce that $x = x$. 

S. Devismes et al (Grenoble I)
Proof

1. \( \overline{1} = 0. \)

According to the definition of the negation, \( x \overline{x} = 0. \) Hence, \( 1 \overline{1} = 0. \)

Since 1 is neutral element for product, we have \( \overline{1} = 0. \)
Proof

1. \( \overline{1} = 0 \).

   According to the definition of the negation, \( x \cdot \overline{x} = 0 \). Hence, \( 1 \cdot \overline{1} = 0 \).
   Since 1 is neutral element for product, we have \( \overline{1} = 0 \).

2. \( \overline{0} = 1 \).
Proof

1. \( \bar{1} = 0 \).
   
   According to the definition of the negation, \( x \cdot \bar{x} = 0 \). Hence, \( 1 \cdot \bar{1} = 0 \).
   
   Since 1 is neutral element for product, we have \( \bar{1} = 0 \).

2. \( \bar{0} = 1 \).
   
   According to the definition of the negation, \( x + \bar{x} = 1 \). Hence, \( 0 + \bar{0} = 1 \).
   
   Since 0 is neutral element for sum, we have \( \bar{0} = 1 \).
Proof

1. $\overline{1} = 0$.

   According to the definition of the negation, $x \overline{x} = 0$. Hence, $1 \overline{1} = 0$. Since 1 is neutral element for product, we have $\overline{1} = 0$.

2. $\overline{0} = 1$.

   According to the definition of the negation, $x + \overline{x} = 1$. Hence, $0 + \overline{0} = 1$. Since 0 is neutral element for sum, we have $\overline{0} = 1$.

3. $\overline{\overline{x}} = x$. 

   According to the properties of negation and commutativity, we have: $x + \overline{x} = 1$, $x \overline{x} = 0$, $x + x = 1$, and $x \overline{x} = 0$. Because of the uniqueness of the negation (property 1.5.3), we deduce that $\overline{\overline{x}} = x$. 

S. Devismes *et al* (Grenoble I)
Proof

1. $\overline{1} = 0$.
   
   According to the definition of the negation, $x.\overline{x} = 0$. Hence, $1.\overline{1} = 0$. Since 1 is neutral element for product, we have $\overline{1} = 0$.

2. $\overline{0} = 1$.
   
   According to the definition of the negation, $x + \overline{x} = 1$. Hence, $0 + \overline{0} = 1$. Since 0 is neutral element for sum, we have $\overline{0} = 1$.

3. $\overline{\overline{x}} = x$.
   
   According to the properties of negation and commutativity, we have: $\overline{x} + x = 1$, $\overline{x}.x = 0$, $\overline{x} + \overline{x} = 1$, and $\overline{x}.\overline{x} = 0$. Because of the uniqueness of the negation (property 1.5.3), we deduce that $x = \overline{x}$. 
Proof

> Product idempotence: $x \cdot x = x$. 

S. Devismes et al (Grenoble I) Logic formulae transformation 18 January 2013 41 / 65
Proof

- Product idempotence: \( x \cdot x = x \).

\[
\begin{align*}
x & = x \cdot 1 \\
& = x(x + \overline{x}) \\
& = x \cdot x + x \cdot \overline{x} \\
& = x \cdot x + 0 \\
& = x \cdot x
\end{align*}
\]
Proof

- Sum idempotence: \( x + x = x \)
Proof

- Sum idempotence: $x + x = x$

\[
\begin{align*}
x & = x + 0 \\
& = x + x \cdot \bar{x} \\
& = (x + x) \cdot (x + \bar{x}) \\
& = (x + x) \cdot 1 \\
& = x + x
\end{align*}
\]
Proof

1 is absorbing element of the sum: \( 1 + x = 1 \).
Proof

- 1 is absorbing element of the sum: \(1 + x = 1\).

We use the sum idempotence.

\[
1 + x = (x + \overline{x}) + x = x + \overline{x} = 1
\]
Proof

- 1 is absorbing element of the sum: \( 1 + x = 1 \).
  
  We use the sum idempotence:
  
  \[
  1 + x = (x + \overline{x}) + x = x + \overline{x} = 1
  \]

- 0 is absorbing element for the product: \( 0 \cdot x = 0 \).
Proof

1 is absorbing element of the sum : $1 + x = 1$.

We use the sum idempotence.

\[
1 + x = (x + \overline{x}) + x \\
= x + \overline{x} \\
= 1
\]

0 is absorbing element for the product : $0 \cdot x = 0$.

We use the product idempotence.

\[
0 \cdot x = (x \cdot \overline{x}) \cdot x \\
= \overline{x} \cdot x \\
= 0
\]
Proof: De Morgan Law: $\overline{xy} = \overline{x} + \overline{y}$
Proof : De Morgan Law : $\overline{xy} = \bar{x} + \bar{y}$

We first show that $xy + (\bar{x} + \bar{y}) = 1$
Proof : De Morgan Law : $xy = \bar{x} + \bar{y}$

We first show that $xy + (\bar{x} + \bar{y}) = 1$

\[
x.y + (\bar{x} + \bar{y}) = (x + \bar{x} + \bar{y}).(y + \bar{x} + \bar{y}) \\
= (1 + \bar{y}).(1 + \bar{x}) \\
= 1.1 \\
= 1
\]
Proof : De Morgan Law : $\overline{xy} = \bar{x} + \bar{y}$

We first show that $xy + (\bar{x} + \bar{y}) = 1$

\[
x.y + (\bar{x} + \bar{y}) = (x + \bar{x} + \bar{y}).(y + \bar{x} + \bar{y})
= (1 + \bar{y}).(1 + \bar{x})
= 1.1
= 1
\]

We also show that $x.y.(\bar{x} + \bar{y}) = 0.$
Proof: De Morgan Law: \( \overline{xy} = \overline{x} + \overline{y} \)

We first show that \( xy + (\overline{x} + \overline{y}) = 1 \)

\[
x.y + (\overline{x} + \overline{y}) = (x + \overline{x} + \overline{y})(y + \overline{x} + \overline{y})
= (1 + \overline{y})(1 + \overline{x})
= 1.1
= 1
\]

We also show that \( x.y.(\overline{x} + \overline{y}) = 0 \)

\[
x.y.(\overline{x} + \overline{y}) = x.y.\overline{x} + x.y.\overline{y}
= 0.y + x.0
= 0 + 0
= 0
\]
Proof: De Morgan Law: $xy = \overline{x} + \overline{y}$

We first show that $xy + (\overline{x} + \overline{y}) = 1$

\[
x.y + (\overline{x} + \overline{y}) = (x + \overline{x} + \overline{y})(y + \overline{x} + \overline{y})
\]
\[
= (1 + \overline{y})(1 + \overline{x})
\]
\[
= 1.1
\]
\[
= 1
\]

We also show that $x.y.(\overline{x} + \overline{y}) = 0$

\[
x.y.(\overline{x} + \overline{y}) = x.y.\overline{x} + x.y.\overline{y}
\]
\[
= 0.y + x.0
\]
\[
= 0 + 0
\]
\[
= 0
\]

Since negation is unique $\overline{x} + \overline{y}$ is the negation of $xy$. 
Proof: De Morgan Law: $\overline{x + y} = \overline{x} \cdot \overline{y}$
Proof: De Morgan Law: $\overline{x + y} = \overline{x} \cdot \overline{y}$

We first show that $(x + y) + \overline{x} \cdot \overline{y} = 1$
Proof : De Morgan Law : $\overline{x + y} = \overline{x} \overline{y}$

We first show that $(x + y) + \overline{x} \overline{y} = 1$

$$(x + y) + \overline{x} \overline{y} = (x + y + \overline{x}).(x + y + \overline{y})$$
$$= (1 + y).(x + 1)$$
$$= 1.1$$
$$= 1$$
Proof: De Morgan Law: \( x + y = \overline{x} \cdot \overline{y} \)

We first show that \((x + y) + \overline{x} \cdot \overline{y} = 1\)

\[
(x + y) + \overline{x} \cdot \overline{y} = (x + y + \overline{x})(x + y + \overline{y})
= (1 + y)(x + 1)
= 1 \cdot 1
= 1
\]

We also show that \((x + y) \cdot \overline{x} \cdot \overline{y} = 0\).

\[
(x + y) \cdot \overline{x} \cdot \overline{y} = (x \cdot \overline{x} \cdot \overline{y}) + (y \cdot \overline{x} \cdot \overline{y})
= (0 \cdot \overline{y}) + (0 \cdot \overline{x})
= 0 + 0
= 0
\]
Proof: De Morgan Law: $\overline{x + y} = \overline{x} \cdot \overline{y}$

We first show that $(x + y) + \overline{x} \cdot \overline{y} = 1$

$$(x + y) + \overline{x} \cdot \overline{y} = (x + y + \overline{x}) \cdot (x + y + \overline{y})$$
$$= (1 + y) \cdot (x + 1)$$
$$= 1.1$$
$$= 1$$

We also show that $(x + y) \cdot \overline{x} \cdot \overline{y} = 0$.

$$(x + y) \cdot \overline{x} \cdot \overline{y} = (x \cdot \overline{x} \cdot \overline{y}) + (y \cdot \overline{x} \cdot \overline{y})$$
$$= (0 \cdot \overline{y}) + (0 \cdot \overline{x})$$
$$= 0 + 0$$
$$= 0$$

From the uniqueness of the negation, we conclude that $\overline{x} \cdot \overline{y}$ is the negation of $(x + y)$.
Definition

Definition 1.5.5

We denote $A^*$ the dual formula of $A$, inductively defined as:

- $x^* = x$,
- $0^* = 1$,
- $1^* = 0$,
- $(\neg A)^* = (\neg A^*)$,
- $(A \lor B)^* = (A^* \land B^*)$,
- $(A \land B)^* = (A^* \lor B^*)$. 
Definition 1.5.5

We denote $A^*$ the dual formula of $A$, inductively defined as:

- $x^* = x$,
- $0^* = 1$,
- $1^* = 0$,
- $(\neg A)^* = (\neg A^*)$,
- $(A \lor B)^* = (A^* \land B^*)$,
- $(A \land B)^* = (A^* \lor B^*)$.

Example 1.5.6

$$(a.(\overline{b} + c))^* =$$
Definition

Definition 1.5.5

We denote \( A^* \) the **dual** formula of \( A \), inductively defined as:

- \( x^* = x \),
- \( 0^* = 1 \),
- \( 1^* = 0 \),
- \( (\neg A)^* = (\neg A^*) \),
- \( (A \lor B)^* = (A^* \land B^*) \),
- \( (A \land B)^* = (A^* \lor B^*) \).

Example 1.5.6

\[
(a.(\bar{b} + c))^* = (a + (\bar{b}.c))
\]
Definition and properties

Theorem 1.5.7
If two formulae are equivalent, their duals are also equivalent.

Corollary 1.5.8
If a formula is valid, its dual is contradictory.

For the proofs, see exercise 29.
Definition 1.5.9: Boolean equality

A formula $A$ is equal to a formula $B$ in a Boolean Algebra:

- $A$ and $B$ are syntactically identical,
- $A$ and $B$ constitute the two members of an axiomes of the Boolean Algebra,
- $B$ equals $A$ (the equality is symmetrical),
- there is a formula $C$ so that $A$ equals $C$ and $C$ equals $B$ (transitivity of the equality),
- there are two formulae $C$ and $D$ so that $C$ equals $D$ and $B$ is obtained by replacing in $A$ an occurrence of $C$ by $D$.

Theorem 1.5.10

If two formulae are equal in a Boolean Algebra, then their duals are also equal.
Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion
Definition 1.6.1: Boolean function

A boolean function is a function whose arguments and the results belong to the set \{0, 1\}.
Definition 1.6.1: Boolean function

A boolean function is a function whose arguments and the results belong to the set \{0, 1\}.

Example 1.6.2

- The function \( f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x \)
**Definition 1.6.1 : Boolean function**

A **boolean function** is a function whose arguments and the results belong to the set \( \{0, 1\} \).

**Example 1.6.2**

- The function \( f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x \)
  is a boolean function.
- The function \( f : \mathbb{N} \rightarrow \mathbb{B} : f(x) = x \mod 2 \)
Definition 1.6.1: Boolean function

A boolean function is a function whose arguments and the results belong to the set \( \{0, 1\} \).

Example 1.6.2

- The function \( f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x \)
  
  is a boolean function.

- The function \( f : \mathbb{N} \rightarrow \mathbb{B} : f(x) = x \mod 2 \)
  
  is not a boolean function.

- The function \( f : \mathbb{B} \rightarrow \mathbb{N} : f(x) = x + 1 \)
Definition 1.6.1: Boolean function

A boolean function is a function whose arguments and the results belong to the set \{0, 1\}.

Example 1.6.2

- The function \( f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x \)
  is a boolean function.

- The function \( f : \mathbb{N} \rightarrow \mathbb{B} : f(x) = x \mod 2 \)
  is not a boolean function.

- The function \( f : \mathbb{B} \rightarrow \mathbb{N} : f(x) = x + 1 \)
  is not a boolean function.

- The function \( f : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B} : f(x, y) = \neg (x \land y) \)
**Definition 1.6.1 : Boolean function**

A **boolean function** is a function whose arguments and the results belong to the set \{0, 1\}.

**Example 1.6.2**

- The function \( f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x \)
  
is a boolean function.

- The function \( f : \mathbb{N} \rightarrow \mathbb{B} : f(x) = \text{x mod } 2 \)
  
is not a boolean function.

- The function \( f : \mathbb{B} \rightarrow \mathbb{N} : f(x) = x + 1 \)
  
is not a boolean function.

- The function \( f : \mathbb{B} \times \mathbb{B} \rightarrow \mathbb{B} : f(x, y) = \neg(x \land y) \)
  
is a boolean function.
Boolean functions and monome sums

**Theorem 1.6.3**

For every variable $x$, we set $x^0 = \overline{x}$ and $x^1 = x$.

Let $f$ be a boolean function of $n$ arguments. This function is represented using $n$ variables $x_1, \ldots, x_n$. Let $A$ the following formula:

$$A = \sum_{f(a_1,\ldots,a_n)=1} x_1^{a_1} \cdots x_n^{a_n}.$$  

$a_i$ are boolean values and $A$ is the sum of the monomes $x_1^{a_1} \cdots x_n^{a_n}$ so that $f(a_1, \ldots, a_n) = 1$. By agreement, if function $f$ always equals 0 then $A = 0$.

For all assignment $\nu$ so that $\nu(x_1) = a_1, \ldots, \nu(x_n) = a_n$, we have $f(a_1, \ldots, a_n) = [A]_\nu$. 

Example 1.6.4

The function $maj$ with 3 arguments equals 1 when at least 2 of its arguments equal 1.

Define the equivalent sum of monomes (theorem 1.6.3)
Example 1.6.4

The function $maj$ with 3 arguments equals 1 when at least 2 of its arguments equal 1.

Define the equivalent sum of monomes (theorem 1.6.3)

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$maj(x_1, x_2, x_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

$maj(x_1, x_2, x_3) = x_1 x_2 \overline{x_3} + x_1 \overline{x_2} x_3 + x_1 x_2 \overline{x_3} + x_1 x_2 x_3$
Let us verify the theorem 1.6.3 on example 1.6.4
Let us verify the theorem 1.6.3 on example 1.6.4

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$maj(x_1, x_2, x_3)$</th>
<th>$\bar{x}_1 \bar{x}_2 x_3$</th>
<th>$x_1 \bar{x}_2 x_3$</th>
<th>$x_1 x_2 \bar{x}_3$</th>
<th>$x_1 x_2 x_3$</th>
<th>$\bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Proof of Theorem 1.6.3

Let \( v \) be any assignment.

Note that for all variable \( x \), \( v(x^a) = 1 \) if and only if \( v(x) = a \).
Proof of Theorem 1.6.3

Let $v$ be any assignment.

Note that for all variable $x$, $v(x^a) = 1$ if and only if $v(x) = a$.

From this remark, we deduce the following property:

$$v(x_1^{a_1} \ldots x_n^{a_n}) = 1 \text{ if and only if } v(x_1) = a_1, \ldots, v(x_n) = a_n.$$  \hfill (1)
Proof of Theorem 1.6.3

Let $v$ be any assignment.

Note that for all variable $x$, $v(x^a) = 1$ if and only if $v(x) = a$. From this remark, we deduce the following property:

$$v(x_1^{a_1} \ldots x_n^{a_n}) = 1 \quad \text{if and only if} \quad v(x_1) = a_1, \ldots, v(x_n) = a_n. \quad (1)$$

Let $a_1, \ldots, a_n$ a list of $n$ boolean values and $v$ an assignment so that $v(x_1) = a_1, \ldots v(x_n) = a_n$. Consider the following two cases:
Proof of Theorem 1.6.3

Let $v$ be any assignment.

Note that for all variable $x$, $v(x^a) = 1$ if and only if $v(x) = a$.

From this remark, we deduce the following property:

$$v(x_1^{a_1} \ldots x_n^{a_n}) = 1 \quad \text{if and only if} \quad v(x_1) = a_1, \ldots, v(x_n) = a_n. \quad (1)$$

Let $a_1, \ldots, a_n$ a list of $n$ boolean values and $v$ an assignment so that $v(x_1) = a_1, \ldots v(x_n) = a_n$.

Consider the following two cases:

1. $f(a_1, \ldots, a_n) = 1$:

2. $f(a_1, \ldots, a_n) = 0$:
Proof of Theorem 1.6.3

Let $v$ be any assignment.

Note that for all variable $x$, $v(x^a) = 1$ if and only if $v(x) = a$. From this remark, we deduce the following property:

$$v(x_1^{a_1} \ldots x_n^{a_n}) = 1 \quad \text{if and only if} \quad v(x_1) = a_1, \ldots, v(x_n) = a_n. \tag{1}$$

Let $a_1, \ldots, a_n$ a list of $n$ boolean values and $v$ an assignment so that $v(x_1) = a_1, \ldots, v(x_n) = a_n$. Consider the following two cases:

1. $f(a_1, \ldots, a_n) = 1$: The monome $x_1^{a_1} \ldots x_n^{a_n}$ is then one of the monomes of $A$. According to (1), we have $v(x_1^{a_1} \ldots x_n^{a_n}) = 1$. Since, according to the definition of $A$, this monome is the element of the sum $A$, we have $[A]_v = 1$.

2. $f(a_1, \ldots, a_n) = 0$:
Proof of Theorem 1.6.3

Let \( v \) be any assignment.

Note that for all variable \( x \), \( v(x^a) = 1 \) if and only if \( v(x) = a \).
From this remark, we deduce the following property :

\[
v(x_1^{a_1} \ldots x_n^{a_n}) = 1 \quad \text{if and only if} \quad v(x_1) = a_1, \ldots, v(x_n) = a_n.
\]  

(1)

Let \( a_1, \ldots, a_n \) a list of \( n \) boolean values and \( v \) an assignment so that \( v(x_1) = a_1, \ldots, v(x_n) = a_n \).
Consider the following two cases :

1. \( f(a_1, \ldots, a_n) = 1 \) : The monome \( x_1^{a_1} \ldots x_n^{a_n} \) is then one of the monomes of \( A \). According to (1), we have \( v(x_1^{a_1} \ldots x_n^{a_n}) = 1 \). Since, according to the definition of \( A \), this monome is the element of the sum \( A \), we have \( [A]_v = 1 \).

2. \( f(a_1, \ldots, a_n) = 0 \) : Let us suppose, by contradiction, that \( [A]_v = 1 \). In that case, there exists a monome of \( A, x_1^{b_1}, \ldots, x_n^{b_n} \), so that \( v(x_1^{b_1} \ldots x_n^{b_n}) = 1 \). According to the definition of \( A \), we have \( f(b_1, \ldots, b_n) = 1 \). Yet, according to (1), we have \( v(x_1^{b_1} \ldots x_n^{b_n}) = 1 \) if and only if \( v(x_1) = b_1, \ldots, v(x_n) = b_n \), thus according to the definition of \( v \), \( a_1 = b_1, \ldots, a_n = b_n \). We therefore obtain a contradiction with \( f(a_1, \ldots, a_n) = 1 \), consequently \( [A]_v = 0 \).
Boolean functions and product of clauses

**Theorem 1.6.5**

For every variable $x$, we set $x^0 = \bar{x}$ and $x^1 = x$.

Let $f$ a boolean function of $n$ arguments. This function is represented using $n$ variables $x_1, \ldots, x_n$. Let $A$ the following formula:

$$A = \prod_{f(a_1, \ldots, a_n) = 0} x_1^{a_1} + \ldots + x_n^{a_n}.$$  

Les $a_i$ are boolean values and $A$ is the product of the clauses $x_1^{a_1} + \ldots + x_n^{a_n}$ so that $f(a_1, \ldots, a_n) = 0$. By agreement, if function $f$ always equals 1 then $A = 1$.

For all assignment $\nu$ so that $\nu(x_1) = a_1, \ldots, \nu(x_n) = a_n$, we have $f(a_1, \ldots, a_n) = [A]_\nu$. 
Proof of theorem 1.6.5

The proof of the theorem is a homework.

Let \( \nu \) any assignment. Note that for every variable \( x \), \( \nu(x^a) = 0 \) if and only if \( \nu(x) \not= a \). From this remark, we deduce the following property:

\[
\nu(x_1^{\overline{a_1}} + \ldots x_n^{\overline{a_n}}) = 0 \iff \nu(x_1) \not= \overline{a_1}, \ldots \nu(x_n) \not= \overline{a_n} \tag{2}
\]
\[
\iff \nu(x_1) = a_1, \ldots \nu(x_n) = a_n. \tag{3}
\]

From the above properties, we deduce as before that \( f(x_1, \ldots x_n) = A \).
Example 1.6.6

The function \textit{maj} of 3 arguments equals 1 if at least 2 of its arguments equal 1.

Define the equivalent product of clauses (theorem 1.6.5)
Example 1.6.6

The function $maj$ of 3 arguments equals 1 if at least 2 of its arguments equal 1.

Define the equivalent product of clauses (theorem 1.6.5)

\[
\begin{array}{ccc|c}
 x_1 & x_2 & x_3 & maj(x_1, x_2, x_3) \\
\hline
 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 0 \\
 0 & 1 & 1 & 1 \\
 1 & 0 & 0 & 0 \\
 1 & 0 & 1 & 1 \\
 1 & 1 & 0 & 1 \\
 1 & 1 & 1 & 1 \\
\end{array}
\]

\[
maj(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)
\]
Let us verify theorem 1.6.5 on the example 1.6.6
Let us verify theorem 1.6.5 on the example 1.6.6

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$\text{maj}(x_1, x_2, x_3)$</th>
<th>$x_1 + x_2 + x_3$</th>
<th>$x_1 + x_2 + \overline{x_3}$</th>
<th>$x_1 + \overline{x_2} + x_3$</th>
<th>$\overline{x_1} + x_2 + x_3$</th>
<th>$(x_1 + x_2 + x_3)$</th>
<th>$(x_1 + x_2 + \overline{x_3})$</th>
<th>$(x_1 + \overline{x_2} + x_3)$</th>
<th>$(\overline{x_1} + x_2 + x_3)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion
**BDDC (Binary Decision Diagram based Calculator)**

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

Plan

Substitution and replacement

Normal forms

Boolean Algebra

Boolean functions

The BDDC tools

Conclusion
Conclusion: Today

- Substitution and replacement
- Normal forms
- Boolean Algebra
- Boolean function
- The BDDC tool
Plan of the Semester

TODAY

- Propositional logic *
- Propositional resolution
- Natural propositional deduction
- First order logic

MIDTERM EXAM

- Basis for the automatic proof
  (≪ first order resolution ≫)
- First order natural deduction

EXAM
Conclusion : Next course

- Resolution
Conclusion

Thank you for your attention.

Questions?

Prove by formula simplification our example

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