

Propositional Resolution

A deductive system

Benjamin Wack

Université Grenoble Alpes

January 2025

Last course

- ▶ Important Equivalences
- ▶ Substitutions and replacement
- ▶ Normal Forms

John, Peter and Mary by simplification

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

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

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

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

with $x \vee (x \wedge y) \equiv x$

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

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

$$\neg j \vee j \vee m \vee p = 1$$

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.

Apply the (unusual) distributivity of disjunction over conjunction:

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

The point 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.

$$\blacktriangleright x \mapsto 0, y \mapsto 0$$

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

Used for modelization (SAT-solvers)

Examples 1.4.8 and 1.4.13

Transformation in **DNF** of the following:

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

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

Transformation in **CNF** of the following:

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

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

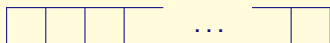
BDDC (*Binary Decision Diagram based Calculator*)

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/>

Example of a SAT modelization

Problem



- ▶ Each square may either contain a token or not.
- ▶ Two tokens can never be neighbours.

Input of the problem: the length n of the grid

Boolean modelization

- ▶ Each square is associated to a boolean variable (true if the square contains a token)
- ▶ For the Dimacs format, we number the squares 1 to n

How to model “At least two squares must contain a token”?

Overview

Boolean Algebra

Boolean functions

Introduction to deductive systems

Resolution and proofs by resolution

Conclusion

Definition 1.5.1

A **Boolean Algebra** is a set with:

- ▶ at least two elements 0 and 1
- ▶ three operations: complement (\bar{x}), sum (+) and product (.)
- ▶ such that :
 1. the sum is associative, commutative, with neutral element 0
 2. the product is associative, commutative, with neutral element 1
 3. the product is distributive over the sum
 4. the sum is distributive over the product
 5. negation laws:
 - ▶ $x + \bar{x} = 1$,
 - ▶ $x \cdot \bar{x} = 0$.

Propositional logic is a Boolean Algebra

The axioms correspond to known equivalences (and can be proven using truth tables).

Another example:

Boolean Algebra	$\mathcal{P}(X)$
1	X
0	\emptyset
\bar{p}	$X - p$
$p + q$	$p \cup q$
$p \cdot q$	$p \cap q$

Example 1.4.10 with Boolean notations

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

Determine whether A is valid.

$$A \equiv (\bar{a} + b).c + a.d$$

$$\begin{aligned} \neg A &\equiv \overline{(\bar{a} + b).c + a.d} \\ &\equiv (\overline{\bar{a} + b + \bar{c}}) . (\bar{a} + \bar{d}) \\ &\equiv (a.\bar{b} + \bar{c}) . (\bar{a} + \bar{d}) \\ &\equiv a.\bar{b}.\bar{a} + a.\bar{b}.\bar{d} + \bar{c}.\bar{a} + \bar{c}.\bar{d} \\ &\equiv a.\bar{b}.\bar{d} + \bar{c}.\bar{a} + \bar{c}.\bar{d} \end{aligned}$$

Properties of a Boolean Algebra

Property 1.5.3

- ▶ For any x , there is exactly one y such that $x + y = 1$ and $xy = 0$.
In other words, the complement is unique.
- ▶
 1. $\bar{1} = 0$
 2. $\bar{0} = 1$
 3. $\bar{\bar{x}} = x$
 4. $x.x = x$
 5. $x + x = x$
 6. $1 + x = 1$
 7. $0.x = 0$
 8. De Morgan laws:
 - ▶ $\overline{xy} = \bar{x} + \bar{y}$
 - ▶ $\overline{x + y} = \bar{x}.\bar{y}$

Moral: some equivalences are fundamental for boolean logic to work, the other ones are just a byproduct.

Proof examples in a Boolean algebra

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

Since $x.\bar{x} = 0$, we have $1.\bar{1} = 0.$

Since 1 is neutral, $\bar{1} = 0.$

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

Ditto : $x + \bar{x} = 1$ hence $0 + \bar{0} = 1.$

Thus $\bar{0} = 1.$

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

By commutativity, $\bar{x} + x = 1$ and $\bar{x}.x = 0.$

Because the complement of \bar{x} is unique, $\bar{\bar{x}} = x.$

Definition 1.6.1: Boolean function

A **boolean function** is a function whose arguments and result belong to the set $\mathbb{B} = \{0, 1\}$.

Example 1.6.2

- ▶ $f : \mathbb{B} \rightarrow \mathbb{B} : f(x) = \neg x$
- ▶ **mais pas** $f : \mathbb{N} \rightarrow \mathbb{B} : f(x) = x \bmod 2$

Thanks to truth tables, we can map any boolean formula to a boolean function.

But is the converse true?

Boolean functions and monomial sums

Theorem 1.6.3

Let $x^0 = \bar{x}$ and $x^1 = x$.

Let f be a boolean function with n arguments, and let:

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

A is the sum of the monomials $x_1^{a_1} \dots x_n^{a_n}$ such that $f(a_1, \dots, a_n) = 1$.

For any assignment v such that $v(x_1) = a_1, \dots, v(x_n) = a_n$, we have $f(a_1, \dots, a_n) = [A]_v$.

Example 1.6.4

The function *maj* with 3 arguments yields 1 when at least 2 of its arguments equal 1.

Define the equivalent sum of monomials (theorem 1.6.3)

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$
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

$$\bar{x}_1 x_2 x_3$$

$$x_1 \bar{x}_2 x_3$$

$$x_1 x_2 \bar{x}_3$$

$$x_1 x_2 x_3$$

$$maj(x_1, x_2, x_3) = \bar{x}_1 x_2 x_3 + x_1 \bar{x}_2 x_3 + x_1 x_2 \bar{x}_3 + x_1 x_2 x_3$$

Let us verify the theorem 1.6.3 on example 1.6.4

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$	$\overline{x_1}x_2x_3$	$x_1\overline{x_2}x_3$	$x_1x_2\overline{x_3}$	$x_1x_2x_3$	$\overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3$
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	1
1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	1
1	1	0	1	0	0	1	0	1
1	1	1	1	0	0	0	1	1

$$maj(x_1, x_2, x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3$$

Boolean functions and product of clauses

Theorem 1.6.5

Let f a boolean function with n arguments, and:

$$A = \prod_{f(a_1, \dots, a_n)=0} x_1^{\bar{a}_1} + \dots + x_n^{\bar{a}_n}.$$

A is the product of the clauses $x_1^{\bar{a}_1} + \dots + x_n^{\bar{a}_n}$ such that $f(a_1, \dots, a_n) = 0$.

For any assignment v such that $v(x_1) = a_1, \dots, v(x_n) = a_n$, we have $f(a_1, \dots, a_n) = [A]_v$.

Example 1.6.6

The function *maj* with 3 arguments yields 1 if at least 2 of its arguments equal 1.

Define the equivalent product of clauses (theorem 1.6.5)

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$
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

$$x_1 + x_2 + x_3$$

$$x_1 + x_2 + \overline{x_3}$$

$$x_1 + \overline{x_2} + x_3$$

$$\overline{x_1} + x_2 + x_3$$

$$maj(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)$$

Let us verify theorem 1.6.5 on the example 1.6.6

x_1	x_2	x_3	$\text{maj}(x_1, x_2, x_3)$	$x_1 + x_2 + x_3$	$x_1 + x_2 + \overline{x_3}$	$x_1 + \overline{x_2} + x_3$	$\overline{x_1} + x_2 + x_3$	$(x_1 + x_2 + x_3)$ $(x_1 + x_2 + \overline{x_3})$ $(x_1 + \overline{x_2} + x_3)$ $(\overline{x_1} + x_2 + x_3)$
0	0	0	0	0	1	1	1	0
0	0	1	0	1	0	1	1	0
0	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0	0
1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

$$\text{maj}(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)$$

Plan of the Semester

- ▶ Propositional logic *
- ▶ Propositional resolution
- ▶ Natural propositional deduction

MIDTERM EXAM

- ▶ First order logic
- ▶ Basis for the automatic proof (“first order resolution”)
- ▶ First order natural deduction

EXAM

Deduction methods

- ▶ Is a formula valid?
- ▶ Is a reasoning correct?

Two methods:

Truth tables and transformations

Problem

- ▶ If the number of variables increases, these methods are very long
- ▶ We only check that the conclusion is consistent with the hypotheses but the underlying reasoning is not given

Example

Using a truth table, to check

$a \Rightarrow b, b \Rightarrow c, c \Rightarrow d, d \Rightarrow e, e \Rightarrow f, f \Rightarrow g, g \Rightarrow h, h \Rightarrow i, i \Rightarrow j \models a \Rightarrow j$
we must test $2^{10} = 1024$ lines.

However, by deduction, this is a correct reasoning:

1. We know that $a \Rightarrow b, b \Rightarrow c \models a \Rightarrow c$.
2. We can apply that step 8 times.
3. We are left with proving that $a \Rightarrow j \models a \Rightarrow j$.
4. By definition, the formula $a \Rightarrow j$ is a consequence of its own.

► Let us formalize a **deductive system**

David Hilbert (1862-1943)

- ▶ Founder of the **formalist** school : mathematics can and should be formalized to be studied.
- ▶ Hilbert's program (1920):
 “*Wir müssen wissen. Wir werden wissen.*”
 as an answer to “*Ignoramus et ignorabimus*”
 - ▶ choose a finite set of axioms to express all of maths
 - ▶ prove it is consistent
 - ▶ design an algorithm that decides whether a proposition can be proved (*Entscheidungsproblem*)
- ▶ *Hilbert-style* deductive systems: axioms such as $\vdash p \Rightarrow (q \Rightarrow p)$
 and a few deduction rules such as
$$\frac{\vdash p \Rightarrow q \quad \vdash p}{\vdash q}$$
- ▶ proofs are thorough but hard to read and to check



Today: propositional resolution

- ▶ only 1 rule: resolution

$$a + \bar{b}, b + c \models a + c$$

for formulas in **CNF (conjunction of clauses)**

- ▶ Formal notion of **proof by resolution**
- ▶ Some properties of resolution

Definitions

Definition 2.1.1

In a clause, the order of the literals and the possible duplicates are meaningless.

Thus, we may say that:

- ▶ A literal is a **member of a clause**.
- ▶ A clause A is **included in a clause B** (or is a **sub-clause of B**) if every literal in A is a member of B .
- ▶ Two clauses are **equal** if they have the same literals.

Example 2.1.2

- ▶ The clauses $p + \bar{q}$, $\bar{q} + p$, and $p + \bar{q} + p$ are equal
- ▶ p is a literal of $\bar{q} + p + r + p$
- ▶ $p + \bar{q} + p$ is a subclause of $\bar{q} + p + r$
- ▶ $(\bar{q} + p + r + p) - p = \bar{q} + r$
- ▶ $(p + p + p) - p = \perp$
- ▶ Adding the literal r to the clause p yields the clause $p + r$
- ▶ Adding the literal p to the clause \perp yields the clause p

Complementary literal

Definition 2.1.4

We note L^c the **complementary literal** of a literal L :

If L is a variable, L^c is the negation of L .

If L is the negation of a variable, L^c is obtained by removing the negation of L .

Example 2.1.5

$$x^c = \bar{x} \text{ and } \bar{x}^c = x.$$

Resolvent

Definition 2.1.6

Let A and B be two clauses.

The clause C is a **resolvent** of A and B iff there exists a literal L such that

$$A = A' + L, \quad B = B' + L^c, \quad C = A' + B'$$

“ C is a resolvent of A and B ” is represented by:

$$\frac{A \quad B}{C}$$

C is generated by A and B .

A and B are the parents of clause C .

Examples with resolution

Example 2.1.7

Give the resolvents of:

- ▶ $p+q+r$ and $p+\bar{q}+r$

$$\frac{p+q+r \quad p+\bar{q}+r}{p+r}$$

- ▶ $p+\bar{q}$ and $\bar{p}+q+r$

$$\frac{p+\bar{q} \quad \bar{p}+q+r}{\bar{p}+p+r} \quad \frac{p+\bar{q} \quad \bar{p}+q+r}{\bar{q}+q+r}$$

- ▶ p and \bar{p}

$$\frac{p \quad \bar{p}}{\perp}$$

Property

Property 2.1.8

If one of the parents of a resolvent is valid, the resolvent is valid or contains the other parent.

Proof.

See exercise 39. □

Example

$$\frac{p + \bar{p} + q \quad \bar{q} + r}{p + \bar{p} + r} \quad \frac{p + \bar{p} + q \quad \bar{p} + r}{\bar{p} + q + r}$$

Definition of a proof

Definition 2.1.11

Let Γ be a set of clauses and C a clause.

A **proof** of C starting from Γ is a list of clauses:

- ▶ where every clause of the proof is a member of Γ
- ▶ or is a resolvent of two clauses already obtained
- ▶ ending with C .

The clause C is **deduced** from Γ (Γ **yields** C , or Γ **proves** C), denoted $\Gamma \vdash C$, if there is a proof of C starting from Γ .

The **size** of a proof is the number of lines in it.

Example

Example 2.1.12

Let Γ be the set of clauses $\bar{p} + q$, $p + \bar{q}$, $\bar{p} + \bar{q}$, $p + q$.

We show that $\Gamma \vdash \perp$:

1	$p + q$	Hyp
2	$p + \bar{q}$	Hyp
3	p	Res 1, 2
4	$\bar{p} + q$	Hyp
5	q	Res 3, 4
6	$\bar{p} + \bar{q}$	Hyp
7	\bar{p}	Res 5, 6
8	\perp	Res 3, 7

Proof tree

Example 2.1.12

Let Γ be the set of clauses $\bar{p} + q, p + \bar{q}, \bar{p} + \bar{q}, p + q$.

We show that $\Gamma \vdash \perp$:

$$\begin{array}{c}
 \frac{\frac{p+q}{p} \quad \frac{p+\bar{q}}{\bar{p}+q}}{q} \quad \bar{p}+q \quad \bar{p}+\bar{q} \quad \frac{\frac{p+q}{p} \quad \frac{p+\bar{q}}{\bar{p}+\bar{q}}}{p}}{\perp}
 \end{array}$$

Monotonicity and Composition

Property 2.1.14

1. **Monotonicity:** If $\Gamma \vdash A$ and if $\Gamma \subseteq \Delta$ then $\Delta \vdash A$
2. **Composition:** If $\Gamma \vdash A$ and $\Gamma \vdash B$ and if C is a resolvent of A and B then $\Gamma \vdash C$.

Proof.

Exercise 38



Today

- ▶ How to use **conjunctive normal forms (CNFs)** to model a problem
- ▶ **Boolean algebra** notations
- ▶ Any **boolean function** can be represented by a (normal) formula
- ▶ A **deductive system** is given by a set of **formal deduction rules** (for instance: resolution)
- ▶ A **proof** is a sequence of applications of these rules starting from **hypotheses**

Next course

- ▶ Correctness and Completeness of the system
- ▶ Davis-Putnam
- ▶ (Complete strategy)

Homework

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.

Transform into clauses the premises and the **negation** of the conclusion.

What can you (or should you) **prove** using resolution ?