Propositional Resolution

A deductive system

Benjamin Wack

Université Grenoble Alpes

January 2024

Last course

- Important Equivalences
- Substitutions and replacement
- Normal Forms

Next week: lecture postponed to Monday 5th at 8AM

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:

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

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.

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

Used for modelization (SAT-solvers)

Examples 1.4.8 and 1.4.13

Transformation in DNF of the following:

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

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

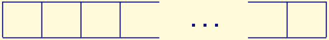
Transformation in CNF of the following:

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

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

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

Definition 1.5.1

A Boolean Algebra is a set with:

- at least two elements 0 and 1
- three operations: complement ($\overline{x}$), 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:
   - $x + \overline{x} = 1$,
   - $x.\overline{x} = 0$.

Propositional logic is a Boolean Algebra

The axioms can be proven using the truth tables.

Another example:

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

Example 1.4.10 with Boolean notations

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

Determine whether A is valid.

$$
\begin{array}{lcl}
A & \equiv & (\overline{a}+b).c+a.d \\
\neg A & \equiv & \overline{(\overline{a}+b).c} \;.\; \overline{a.d} \\
 & \equiv & (\overline{\overline{a}+b}+\overline{c}) \;.\; (\overline{a}+\overline{d}) \\
 & \equiv & (a.\overline{b}+\overline{c}) \;.\; (\overline{a}+\overline{d}) \\
 & \equiv & a.\overline{b}.\overline{a}+a.\overline{b}.\overline{d}+\overline{c}.\overline{a}+\overline{c}.\overline{d} \\
 & \equiv & a.\overline{b}.\overline{d}+\overline{c}.\overline{a}+\overline{c}.\overline{d}
\end{array}
$$

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. $\overline{1} = 0$
  2. $\overline{0} = 1$
  3. $\overline{\overline{x}} = x$
  4. $x.x = x$
  5. $x + x = x$
  6. $1 + x = 1$
  7. $0.x = 0$
  8. De Morgan laws:
     - $\overline{xy} = \bar{x} + \bar{y}$
     - $\overline{x + y} = \bar{x}.\bar{y}$

Proof

1. $\overline{1} = 0$.
   Since $x.\overline{x} = 0$, we have $1.\overline{1} = 0$.
   Since 1 is neutral, $\overline{1} = 0$.
2. $\overline{0} = 1$.
   Ditto : $x + \overline{x} = 1$ hence $0 + \overline{0} = 1$.
   Thus $\overline{0} = 1$.
3. $\overline{\overline{x}} = x$.
   By commutativity, $\overline{x} + x = 1$ and $\overline{x}.x = 0$.
   Because the complement of $\bar{x}$ is unique, $\overline{\overline{x}} = x$.

Proof

4. Product idempotence: $x.x = x$.

$$
\begin{aligned}
x &= x.1 \\
&= x.(x + \overline{x}) \\
&= x.x + x.\overline{x} \\
&= x.x + 0 \\
&= x.x
\end{aligned}
$$

5. Sum idempotence: $x + x = x$

   Ditto, starting from $x = x + 0$.

Proof

6. 1 is an absorbing element of the sum: $1 + x = 1$.
   We use sum idempotence.

$$\begin{aligned} 1 + x &= (x + \overline{x}) + x \\ &= x + \overline{x} \\ &= 1 \end{aligned}$$

7. 0 is an absorbing element for the product: $0.x = 0$.

$$\boxed{\text{Ditto from } 0.x = (x.\overline{x}).x}$$

Proof: De Morgan Law: $\overline{xy} = \bar{x} + \bar{y}$

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

$$\begin{aligned} x.y + (\overline{x} + \overline{y}) &= (x + \overline{x} + \overline{y}).(y + \overline{x} + \overline{y}) \\ &= (1 + \overline{y}).(1 + \overline{x}) \\ &= 1.1 \\ &= 1 \end{aligned}$$

Similarly $x.y.(\overline{x} + \overline{y}) = 0$.
Since negation is unique $\overline{x} + \overline{y}$ is the negation of xy.

Similarly we can prove that $\overline{x + y} = \overline{x}.\overline{y}$ by switching the uses of . and $+$ in this demonstration.

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

Which of these functions are boolean ?

- $f : \mathbb{B} \to \mathbb{B} : f(x) = \neg x$
- $f : \mathbb{N} \to \mathbb{B} : f(x) = x \bmod 2$
- $f : \mathbb{B} \to \mathbb{N} : f(x) = x + 1$
- $f : \mathbb{B} \times \mathbb{B} \to \mathbb{B} : f(x, y) = \neg(x \wedge y)$

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	$\overline{x_1}x_2x_3$
1	0	0	0	
1	0	1	1	$x_1\overline{x_2}x_3$
1	1	0	1	$x_1x_2\overline{x_3}$
1	1	1	1	$x_1x_2x_3$

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

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

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

Boolean functions and product of clauses

Theorem 1.6.5

Let f a boolean function with n arguments, and:

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

A is the product of the clauses $x_1^{\overline{a_1}} + \ldots + x_n^{\overline{a_n}}$ such that $f(a_1, \ldots, a_n) = 0$.

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

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	$x_1 + x_2 + x_3$
0	0	1	0	$x_1 + x_2 + \overline{x_3}$
0	1	0	0	$x_1 + \overline{x_2} + x_3$
0	1	1	1	
1	0	0	0	$\overline{x_1} + x_2 + x_3$
1	0	1	1	
1	1	0	1	
1	1	1	1	

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

Let us verify theorem 1.6.5 on the example 1.6.6

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

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

`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/`

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.

Or, by deduction, this is a correct reasoning:

1. By transitivity of the implication, $a \Rightarrow j \models a \Rightarrow j$.
2. 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 $\dfrac{\vdash p \Rightarrow q \qquad \vdash p}{\vdash q}$
- proofs are thorough but hard to read and to check

Today: propositional resolution

- only 1 rule: resolution

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

  for formulas in CNF (conjunction of clauses)
- Formal notion of proof by resolution
- Some properties of resolution

Definitions

Definition 2.1.1

A clause is identified to the set of its literals (unordered, no duplicates), so 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).
- Two clauses are equal if they have the same literals.

Example 2.1.2

- The clauses $p+\overline{q}$, $\overline{q}+p$, and $p+\overline{q}+p$ are equal
- $p \in \overline{q}+p+r+p$
- $p+\overline{q} \subseteq \overline{q}+p+r+p$
- $\overline{q}+p+r+p - p = \overline{q}+r$
- $p+p+p - p = \bot$
- Adding the literal r to the clause p yields the clause $p+r$
- Adding the literal p to the clause $\bot$ 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

$x^c = \overline{x}$ and $\overline{x}^c = x$.

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

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

"C is a resolvent of A and B" is represented by:

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

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 $p+\overline{q}+r$

$$\frac{p+q+r \qquad p+\overline{q}+r}{p+r}$$

- $p+\overline{q}$ and $\overline{p}+q+r$

$$\frac{p+\overline{q} \qquad \overline{p}+q+r}{\overline{p}+p+r} \qquad\qquad \frac{p+\overline{q} \qquad \overline{p}+q+r}{\overline{q}+q+r}$$

- p and $\overline{p}$

$$\frac{p \qquad \overline{p}}{\perp}$$

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

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

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 $\overline{p}+q$, $p+\overline{q}$, $\overline{p}+\overline{q}$, $p+q$.

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

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

Proof tree

Example 2.1.12

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

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

$$\dfrac{\dfrac{\dfrac{\dfrac{p+q \quad p+\overline{q}}{p} \quad \overline{p}+q}{q} \quad \overline{p}+\overline{q}}{\overline{p}} \quad \dfrac{p+q \quad p+\overline{q}}{p}}{\bot}$$

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

- **Important equivalences** correspond to computation rules in a Boolean algebra
- Any boolean function can be represented by a (normal) formula

- A deductive system is given by a set of **formal rules**
- A proof is a sequence of applications of these rules starting from **hypotheses**.

Next course

- Correctness and Completeness of the system
- Comprehensive strategy
- Davis-Putnam

Next week: lecture postponed to Monday 5th at 8AM

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 ?