Propositional Resolution

First part

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

Université Joseph Fourier, Grenoble I

January 23, 2015

Last course

- Substitutions and replacement
- Normal Forms
- Boolean Algebra
- Boolean functions
- The BDDC tools

John, Peter and Mary by simplification

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

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

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

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

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

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

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

$$\neg j \vee j \vee m \vee p \equiv \top$$

Overview

Deduction methods

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

Two methods :

The truth tables and transformations

Problem

If the number of variables increases, these methods are very long

Example

By a truth table, to verify
$a \Rightarrow b, b \Rightarrow c, c \Rightarrow d, d \Rightarrow e, e \Rightarrow f, f \Rightarrow g, g \Rightarrow h, h \Rightarrow i, i \Rightarrow j \models a \Rightarrow j$
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.

Today

- Formalisation of a deductive system (with 1 rule)
- How to prove a formula by resolution
- Correctness of a deductive system
- Completeness of a deductive system
- Some properties of resolution

Intuition

Formulas are put into CNF (conjunction of clauses)

$$a \vee \neg b, b \vee c \models a \vee c$$

Can be seen as transitivity of implication

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

Definitions

Definition 2.1.1

- A literal is a member of a clause, if it is a member of the set of literals of the clause.
- A clause A is included in a clause B, if all literals of clause A are members of clause B. In this case, A is a sub-clause of B.
- Two clauses are equal if they have the same set of literals.

Example 2.1.2

Notation

$s(A)$ the set of literals of the clause A.
By convention $\perp$ is the empty clause and $s(\perp) = \emptyset$.

Example 2.1.3

$s(\neg q \vee p \vee r \vee p \vee \neg p) =$

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

$x^c = \neg x$ and $\neg x^c = x$.

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 $L \in s(A), L^c \in s(B), s(C) = (s(A) - \{L\}) \cup (s(B) - \{L^c\})$.

"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 the clause C.

Examples with resolution

Example 2.1.7

Give the resolvents of :

- $p \vee q \vee r$ and $p \vee \neg q \vee r$
- $p \vee \neg q$ and $\neg p \vee q \vee r$
- p and $\neg p$

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 40. □

Problem with $\vee$

Given two clauses A and B, the formula $A \vee B$ is not a clause if one of the two operands of the disjunction is the empty clause.

Example : $\perp \vee p$ is not a clause.

Solution : $\tilde{\vee}$

Definition 2.1.9

Let C and D be two clauses.

We denote $C \tilde{\vee} D$ the following clause :

- If $C = \perp$ then $C \tilde{\vee} D = D$,
- else if $D = \perp$ then $C \tilde{\vee} D = C$ else $C \tilde{\vee} D = C \vee D$.

Adding a literal L to the clause C, is building $C \tilde{\vee} L$.

Resolvent : another definition

Definition 2.1.10

Let A and B be two clauses.

The clause C is a resolvent of A and B if and only if there is a literal L such that :

- L is a member of the clause A, L^c is a member of the clause B
- C equals a clause $A' \tilde{\vee} B'$ where $A' = A - \{L\}$ is obtained by removing L from A and $B' = B - \{L^c\}$ is obtained by removing L^c from B.

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 ending by C. Every clause of the proof is a member of Γ or is a resolvent of the two clauses already obtained.

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.

Example

Example 2.1.12

Let Γ be the set of clauses $\neg p \vee q$, $p \vee \neg q$, $\neg p \vee \neg q$, $p \vee q$.

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

Proof tree

Example 2.1.12

Let Γ be the set of clauses $\neg p \vee q$, $p \vee \neg q$, $\neg p \vee \neg q$, $p \vee q$.

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

Definition 2.1.13

Proof length

A proof P of C starting from a set of clauses Γ is of length n if it contains n lines.

Monotony and Composition

Property 2.1.14

Let Γ, Δ be two sets of clauses and A, B be two clauses.

1. Monotony of deduction : If $\Gamma \vdash A$ and if Γ is included in Δ then $\Delta \vdash A$
2. Composition of deductions : If $\Gamma \vdash A$, $\Gamma \vdash B$ and if C is a resolvent of A and B then $\Gamma \vdash C$.

Proof.

Exercise 39 □

Definition

The correctness of a logic system states that all proofs obtained in this system are ≪ correct ≫.

Correctness of the resolution rule

Theorem 2.1.15

If C is a resolvent of A and B then $A, B \models C$.

Proof.

If C is a resolvent of A and B, then there is a literal L so that $L \in s(A), L^c \in s(B), s(C) = (s(A) - \{L\}) \cup (s(B) - \{L^c\})$.

Let v a model truth assignment of A and B. We have $[A]_v = 1$ and $[B]_v = 1$

Let us show that $[C]_v = 1$.

$\square$

Correctness of the deduction

Theorem 2.1.16

Let Γ a set of clauses and C a clause. If $\Gamma \vdash C$ then $\Gamma \models C$.

Proof.

Suppose that $\Gamma \vdash C$. There is a proof P of C starting from Γ. Suppose that for all proof of D starting from Γ, shorter than P, we have $\Gamma \models D$.

Let us show that $\Gamma \models C$. There are two possible cases :

$\square$

Definition

Completeness for the refutation is the following property : If $\Gamma \models \bot$ then $\Gamma \vdash \bot$.

We prove this result for finite Γ.

$\Gamma[L := \top]$

Definition 2.1.18

Let Γ be a set of clauses and L a literal.

$\Gamma[L := \top]$ is the set of clauses obtained by deleting the clauses for which L is a member and by removing L^c from the other clauses.

We define $\Gamma[L := \bot]$ as $\Gamma[L^c := \top]$.

Examples

Example 2.1.19

Let Γ be the set of clauses $\neg p \vee q$, $\neg q \vee r$, $p \vee q$, $p \vee r$. We have :

- $\Gamma[p := \top] =$
- $\Gamma[p := \bot] =$

Let us observe that :

- $(\neg\top \vee q) \wedge (\neg q \vee r) \wedge (\top \vee q) \wedge (\top \vee r) \equiv$
- $(\neg\bot \vee q) \wedge (\neg q \vee r) \wedge (\bot \vee q) \wedge (\bot \vee r) \equiv$

Notation and definition

Intuitively, $v[L \mapsto 1]$ is the truth assignment giving to L the value 1, to L^c the value 0 and which does not change the value of the other literals.

Definition 2.1.20

Let a truth assignment v, the truth assignment $v[L \mapsto 1]$ is an assignment identical to v except possibly for x, the variable of L. If $L = x$ then $v[L \mapsto 1](x) = 1$, if $L = \neg x$ then $v[L \mapsto 1](x) = 0$.

We define $v[L \mapsto 0]$ as $v[L^c \mapsto 1]$.

Property of $\Gamma[L := x]$

Property 2.1.21

Let Γ a set of clauses and L a literal. Γ has a model if and only if $\Gamma[L := \top]$ or $\Gamma[L := \bot]$ has a model.

Proof.

Let v be a truth assignment.

$\Rightarrow$ The truth assignment v is a model of Γ.

$\Leftarrow$ $\Gamma[L := \top]$ or $\Gamma[L := \bot]$ has a model.

$\square$

First case : v is model of Γ

1. Suppose that v gives to L the value 1 and let us show that v is a model of $\Gamma[L := \top]$.
   Let C a clause of $\Gamma[L := \top]$. There is in Γ a clause C' such that C is obtained by removing L^c from C'. Since v is model of Γ, v is model of C' hence of a literal which is not L^c (since L^c equals 0 in this truth assignment). Consequently, v is model of C. Since C is any clause of $\Gamma[L := \top]$, v is model of $\Gamma[L := \top]$.
2. Suppose that v gives to L the value 0. We get back to the previous case by exchanging L and L^c and we show that v is model of $\Gamma[L := \bot]$.

Second case : $\Gamma[L := \top]$ or $\Gamma[L := \bot]$ has a model

Let C be a clause of Γ.

1. Suppose that the truth assignment v is model of $\Gamma[L := \top]$. Let us show that $v[L := \top]$ is model of Γ. Let C be a clause of Γ.
   1.1 Suppose that L is a literal of C, then $v[L := \top]$ is model of C since this truth assignment gives to L the value 1.
   1.2 Suppose that L is not a literal of C. Then there is a clause C' member of $\Gamma[L := \top]$ such that C' is obtained by removing L^c from C. The variable of L is not a variable of C'. Consequently v and $v[L := \top]$ give the same value to C'. Since v is model of $\Gamma[L := \top]$, v is model of C' therefore $v[L := \top]$ is model of C'. Since C' is included in C, $v[L := \top]$ is model of C.

   Since C is any clause of Γ, $v[L := \top]$ is model of Γ.
2. Suppose the truth assignment v is model of $\Gamma[L := \bot]$. By an analogous proof, we show that $v[L := \bot]$ is model of Γ.

Lemma 2.1.22

Lemma 2.1.22

Let Γ a set of clauses, C a clause and L a literal. If $\Gamma[L := \top] \vdash C$ then $\Gamma \vdash C$ or $\Gamma \vdash C \tilde{\vee} L^c$.

Proof.

Starting from a proof of C starting from $\Gamma[L := \top]$, we obtain a proof of C or of $C \tilde{\vee} L^c$ starting from Γ by adding a literal L^c to the clauses where it has been removed from.

Let us formalise this tentative proof. Suppose that $\Gamma[L := \top] \vdash C$. There is a proof P of C starting from $\Gamma[L := \top]$. Suppose that for all proof of D starting from $\Gamma[L := \top]$, shorter than P, we have $\Gamma \vdash D$ or $\Gamma \vdash D \tilde{\vee} L^c$. There are two possible cases :

1. C is a member of $\Gamma[L := \top]$.
2. C is resolvent of 2 clauses A and B preceding C in the proof P.

□

First case : C is a member of $\Gamma[L := \top]$

Let us examine those two cases.

1. Suppose $s(C') = s(C)$.
2. Suppose $s(C') = s(C) \cup \{L^c\}$.

Second case : C is resolvent of 2 clauses A and B preceding C in the proof P

Hence by induction hypothesis :

- $\Gamma \vdash A$ or $\Gamma \vdash A \tilde{\vee} L^c$
- $\Gamma \vdash B$ or $\Gamma \vdash B \tilde{\vee} L^c$

Which results in 4 cases to examine.

1. Suppose $\Gamma \vdash A$ and $\Gamma \vdash B$.
2. Suppose $\Gamma \vdash A$ and $\Gamma \vdash B \tilde{\vee} L^c$. Since C is resolvent of A and B, there is M such that $M \in A$ and $M^c \in B$ and $s(C) = (s(A) - \{M\}) \cup (s(B) - \{M^c\})$. No clause of $\Gamma[L := \top]$ involves the literal L^c. Hence B which deducts from it, does not contain the literal L^c (see exercise 41) and consequently $L^c \neq M^c$. Consequently $(s(B) - \{M^c\}) \cup \{L^c\} = (s(B) \cup \{L^c\}) - \{M^c\} = (s(B \tilde{\vee} L^c) - \{M^c\})$. We therefore have $s(C \tilde{\vee} L^c) = (s(A) - \{M\}) \cup (s(B) - \{M^c\}) \cup \{L^c\} = (s(A) - \{M\}) \cup (s(B \tilde{\vee} L^c) - \{M^c\})$ And consequently $C \tilde{\vee} L^c$ is a resolvent of A and $B \tilde{\vee} L^c$. Hence according to property 2.1.14, $\Gamma \vdash C \tilde{\vee} L^c$.
3. Suppose $\Gamma \vdash A \tilde{\vee} L^c$ and $\Gamma \vdash B$, by exchanging in the above case the roles of A and B, we obtain $\Gamma \vdash C \tilde{\vee} L^c$.
4. Suppose $\Gamma \vdash A \tilde{\vee} L^c$ and $\Gamma \vdash B \tilde{\vee} L^c$, as above we obtain $\Gamma \vdash C \tilde{\vee} L^c$.

Consequently in the four cases, we have $\Gamma \vdash C$ or $\Gamma \vdash C \tilde{\vee} L^c$.

Lemma 2.1.23

Lemma 2.1.23

Let Γ a set of clauses, C a clause and L a literal.

If $\Gamma[L := \bot] \vdash C$ then $\Gamma \vdash C$ or $\Gamma \vdash C \tilde{\vee} L$.

Proof.

Suppose $\Gamma[L := \bot] \vdash C$. Since $\Gamma[L := \bot] = \Gamma[L^c := \top]$ and since $L^{cc} = L$, according to lemma 2.1.22 we have $\Gamma \vdash C$ or $\Gamma \vdash C \tilde{\vee} L$. □

Completeness of propositional resolution

Theorem 2.1.24

Let Γ a finite set of clauses. If Γ is unsatisfiable then $\Gamma \vdash \bot$.

Proof.

Suppose that Γ is unsatisfiable.

We show that $\Gamma \vdash \bot$ by induction on the number of variables of Γ.

Hypothesis : Suppose that for all set Δ of unsatisfiable clauses with less than n variables, we have $\Delta \vdash \bot$.

Let Γ unsatisfiable with n variables. Let us show that $\Gamma \vdash \bot$. We distinguish two cases depending on whether n is null or not.

□

The base case (basis)

Suppose that n is null.

Hence $\Gamma = \emptyset$ or $\Gamma = \{\bot\}$. The first case is impossible, since the empty set is valid (any truth assignment is a model of it). Hence $\Gamma = \{\bot\}$ and consequently $\Gamma \vdash \bot$.

Inductive step

Suppose that n is not null.

Let x a variable appearing in Γ. According to the property 2.1.21, $\Gamma[x := \bot]$ and $\Gamma[x := \top]$ are unsatisfiable.
Since the variable x does not appear in these two sets of clauses, the induction hypothesis applies, hence : $\Gamma[x := \bot] \vdash \bot$ and $\Gamma[x := \top] \vdash \bot$. From lemmas 2.1.22 and 2.1.23, we deduce either $\Gamma \vdash \bot$, or $\Gamma \vdash \neg x$ and $\Gamma \vdash x$. In the first case, the proof is finished. In the second case, since $\bot$ is a resolvent of $\neg x$ and x, we also have $\Gamma \vdash \bot$.

Conclusion

Corollary 2.1.25

Let Γ a finite set of clauses. Γ is unsatisfiable if and only if $\Gamma \vdash \bot$.

Conclusion : Today

- Formalisation of a deductive system
- Correctness of the system
- Completeness of the system

Conclusion : Next course

- Comprehensive strategy
- Davis-Putnam

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.

Prove, using resolution, that we can derive the conclusion C from the premises H1, H2, H3.

Hint : Transform into clauses the premises and the negation of the conclusion.

Conclusion

Thank you for your attention.

Questions ?