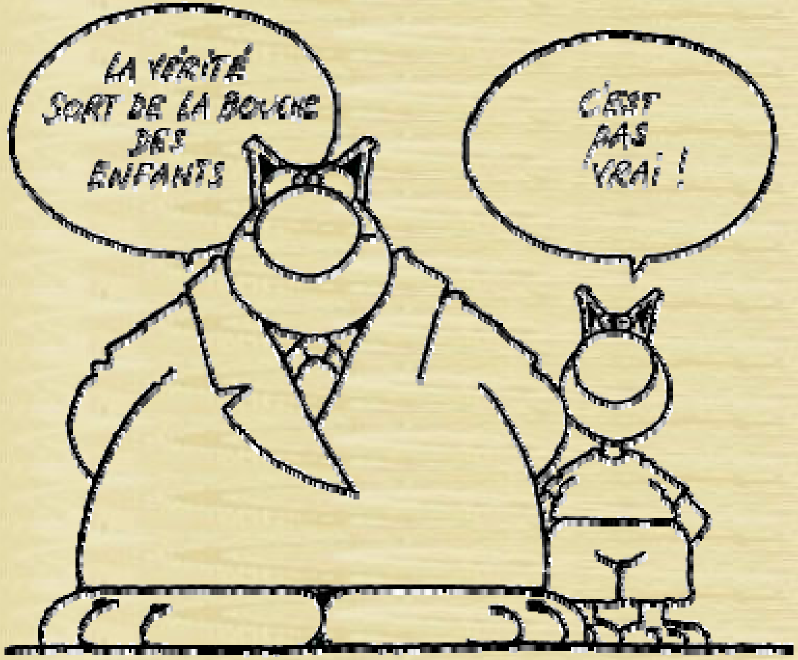
DATAAI-900

Propositional Logic

Jean-Louis Dessalles

Dep. InfRes



Voir aussi la bande dessinée:
http://www.logicomix.com/fr/

Logic is essential for

- Computer science
- Automated theorem proving
- Proofs of programs
- AI and reasoning
  - Argumentation & high-level NLP
- Electronics
- Database management
- Knowledge Representation & semantic Web
- . . .

Logic

- Human adventure
  - Cognition
  - Proof – automated proof
- Contradiction
  - Anomaly detection
  - Explanation (XAI)
- Relevance
  - No continuity
  - Reason *vs.* guess
- Basic in many curriculums

History

http://individual.utoronto.ca/pking/miscellaneous/history-of-logic.pdf

- Ancient greeks
  - Stoics
  - Aristotle
    - Syllogism
    - Argumentation
- Medieval logic
  - William of Ockham (1288-1348)
    - de Morgan's laws
    - Ternary logic
- Traditional logic
  - Port Royal's logic
    - Antoine Arnauld & Pierre Nicole (1662)
      - Logic of propositions
- Modern Logic
  - Descartes, Leibniz
  - George Boole (1848)
  - Gottlob Frege
    - *Bregriffschrift* (1879)
      - *Quantification*
  - Charles Peirce
  - Guiseppe Peano
    - Logical axiomatization of artihmetics
  - Bertand Russell & Alfred N. Whitehead (1925)
    - Logical axiomatization of mathematics

All Bankers are Athletes,
No Consultant is a Banker.
So…

Some athletes aren’t consultants.

Syntax

Alphabet

symboles propositionnels : p1 , p2 , ...

connecteurs

à 0 place : T, $\perp$ (constantes)

à 1 place : $\neg$

à 2 places : $\wedge \vee \supset \subset \uparrow \downarrow \not\subset \not\supset \equiv$

Formule Atomique

si F est une constante, alors F $\in$ ***A***

si F est un symbole propositionnel, alors F $\in$ ***A***

Formule Propositionnelle

si F est une formule atomique, alors F $\in$ ***F***

si F $\in$ ***F***, alors $\neg F \in$ ***F***

si $\bullet$ est un connecteur à deux places, si $F_1 \in$ ***F*** et $F_2 \in$ ***F***, alors $(F_1 \bullet F_2) \in$ ***F***

F est le plus petit ensemble possédant ces propriétés

Truth values

Space of *truth values*: $T = \{\boldsymbol{T}, \boldsymbol{F}\}$ (propositional attitudes)

	not
T	F
F	T

		and	or	imp	impinv	nand	nor	nimp	nimpinv	equ	nequ
T	**T**	T	T	T	T	F	F	F	F	T	F
T	**F**	F	T	F	T	T	F	T	F	F	T
F	**T**	F	T	T	F	T	F	F	T	F	T
F	**F**	F	F	T	T	T	T	F	F	T	F

		T	***F***	*"degenerate" connectives*			
T	**T**	T	F	T	T	F	F
T	**F**	T	F	T	F	F	T
F	**T**	T	F	F	T	T	F
F	**F**	T	F	F	F	T	T

Truth tables

X	Y	$(X \wedge Y)$
T	T	T
T	F	F
F	T	F
F	F	F

X	Y	$(X \vee Y)$
T	T	T
T	F	T
F	T	T
F	F	F

X	Y	$(X \supset Y)$
T	T	T
T	F	F
F	T	T
F	F	T

X	Y	$(X \equiv Y)$
T	T	T
T	F	F
F	T	F
F	F	T

valuations

By extension, one draws truth tables for propositional formula.

"Semantics" (in fact, truth values)

Boolean valuation

$\nu : F \to T$

$\nu\,(F) \in \{v, f\}$

$\nu\,(T) = v$

$\nu\,(\bot) = f$

$\nu\,(\neg F) = \text{non}\ \nu\,(F)$

$\nu\,((F_1 \bullet F_2)) = \nu\,(F_1)\ \blacklozenge\ \nu\,(F_2)$

Syntactic connective $\bullet$	Semantic connective $\blacklozenge$
$\neg$	Not
$\wedge$	And
$\vee$	Or
$\supset$	$\Rightarrow$
$\subset$	$\Leftarrow$
$\uparrow$	Nand
$\downarrow$	Nor
$\not\supset$	$\not\Rightarrow$
$\not\subset$	$\not\Leftarrow$

Tautologies and satisfiability

•A propositional formula X is a *tautology*
if $v(X) = \boldsymbol{T}$ for any Boolean valuation v.

One can see that X is a tautology iff ($X \equiv \boldsymbol{T}$) is a tautology.

•A set S of propositional formulas is *satisfiable*
if some valuation v_0 maps every member of S to T:
$v_0(X) = \boldsymbol{T}$ for any X of S.

X is a tautology if and only if $\{\neg X\}$ is not satisfiable.

Show that $(\neg(X \wedge Y) \equiv (\neg X \vee \neg Y))$ is a tautology.

Show that X is a tautology iff
$(X \equiv \mathrm{T})$ is a tautology, and iff
$(\mathrm{T} \supset X)$ is a tautology.

Distributivity

$(P \wedge (Q \vee R) \equiv ((P \wedge Q) \vee (P \wedge R))$ is a tautology.

$(P \vee (Q \wedge R) \equiv ((P \vee Q) \wedge (P \vee R))$ is a tautology.

De Morgan's laws

$(\neg(P \vee Q) \equiv (\neg P \wedge \neg Q))$ is a tautology.

$(\neg(P \wedge Q) \equiv (\neg P \vee \neg Q))$ is a tautology.

Logical consequence

$S \models X$ (*logical consequence*)
if a valuation assigns the value ***T*** to all element of *S*, then it will assign ***T*** to *X*.

$\models X$
X is a tautology.

Show that if $S \models X$, then $S \cup \{\neg X\}$ is not satisfiable.
Show the reciprocal (refutation)

(*ex falso quodlibet sequitur*).
Show that if $A, \neg A \in S$, then for any X : $S \models X$.
Conversely, if for any X : $S \models X$, then
show that S is not satisfiable.

(monotony).
Show that if $S \models X$ then $S \cup \{Y\} \models X$

(deduction).
Show that $S \cup \{X\} \models Y$ iff $S \models (X \supset Y)$

Automated theorem proving

- Problem: Prove that a given formula is a tautotoly
- Means:
  - Modify the formula while preserving its truth value (possibly by embedding it into a larger language)
  - until the truth value can easily be determined

Replacement Theorem

$F(P)$ is a propositional formula with zero, one or several occurrences of symbol P

Replacement Theorem

if $(X \equiv Y)$ is a tautology, then $(F(X) \equiv F(Y))$ is a tautology as well.

$(\neg\neg X \equiv X)$ is a tautology.

$(\neg\neg X \supset p)$ et $(X \supset p)$ have same "semantics"

(*modus ponens*)
Show that Y is a tautology
if X and $(X \supset Y)$ are tautologies

Generalized Disjonction, Generalized Conjunction

$X_1, X_2, \ldots, X_n$ are propositional formulas. We define new formulas (that do not belong to the propositional language):

$[X_1, X_2, \ldots, X_n]$ is the generalized disjunction of $X_1, X_2, \ldots, X_n$.

$< X_1, X_2, \ldots, X_n >$ is the generalized conjunction of $X_1, X_2, \ldots, X_n$.

If ν is a Boolean valuation:

$\nu\,([X_1, X_2, \ldots, X_n]) = \mathbf{F}$ iff $\nu(X_i) = \mathbf{F}$ for all i.

$\nu\,(<X_1, X_2, \ldots, X_n >) = \mathbf{T}$ iff $\nu(X_i) = \mathbf{T}$ for all i.

$\nu\,([\ ]) = \mathbf{F}$ $\qquad([\ ] \equiv \bot)$

$\nu\,(< >) = \mathbf{T}$ $\qquad(< > \equiv \mathrm{T})$

Conjunctive Normal form

The *cunjunctive normal form* (CNF) rewrites any propositional formula as a conjunction of *clauses*;
each of these clauses is a generalized disjunction of propositional symbols possibly with negation.

A **clause** is noted [*a, b, c*].
A conjunction of clauses is noted $< C_1, C_2, C_3 >$.
A propositional formula is *in* ***CNF*** if it is written as a conjunction of clauses :
$< C_1, C_2, \dots C_n >$ where each C_i is a clause.

$v([X_1, X_2, \dots X_n]) = \boldsymbol{F}$ iff $v(X_i) = \boldsymbol{F}$ for all i
$v(< C_1, C_2, \dots C_m >) = \boldsymbol{T}$ iff $v(C_i) = \boldsymbol{T}$ for all i

$v([\,]) = \boldsymbol{F}$, $v(< >) = \boldsymbol{T}$.

Formes normales disjonctives, conjonctives

A propositional formula is in *normal disjunctive form* if it is written as a generalized disjunction [C_1, C_2, …, C_n] where each C_i is a generalized conjunction.

A propositional formula is in *normal conjunctive form* if it is written as a generalized conjunction <D_1, D_2, …, D_n> where each D_i is a clause (generalized disjunction).

There are algorithms that convert any propositional formula *X* into a normal (conjunctive or disjunctive) formula *Y*, so that (X ≡Y) is a tautology.

Remplacing primitive connectives

For any Boolean valuation ν,

- $\nu(\alpha) = (\ \nu\ (\alpha_1)$ and $\nu\ (\alpha_2)\ )$
- $\nu\ (\beta) = (\ \nu\ (\beta_1)$ ou $\nu\ (\beta_2)\ )$.

For any α or β formula, $\alpha \equiv (\alpha_1 \wedge \alpha_2)$, $\beta \equiv (\beta_1 \vee \beta_2)$ are tautologies.

α	α_1	α_2
$(X \wedge Y)$	X	Y
$\neg(X \vee Y)$	$\neg X$	$\neg Y$
$\neg(X \supset Y)$	X	$\neg Y$
$\neg(X \subset Y)$	$\neg X$	Y
$\neg(X \uparrow Y)$	X	Y
$(X \downarrow Y)$	$\neg X$	$\neg Y$
$(X \not\supset Y)$	X	$\neg Y$
$(X \not\subset Y)$	$\neg X$	Y

β	β_1	β_2
$\neg(X \wedge Y)$	$\neg X$	$\neg Y$
$(X \vee Y)$	X	Y
$(X \supset Y)$	$\neg X$	Y
$(X \subset Y)$	X	$\neg Y$
$(X \uparrow Y)$	$\neg X$	$\neg Y$
$\neg(X \downarrow Y)$	X	Y
$\neg(X \not\supset Y)$	$\neg X$	Y
$\neg(X \not\subset Y)$	X	$\neg Y$

Conjunctive Normal form

The algorithm that converts a propositional formula into CNF proceeds step by step (based on the replacement theorem). Basic transitions are:

replace < ... [... β ...] ... > by < ... [... β_1, β_2 ...] ... >

replace < ... [... α ...] ... > by < ... [... α_1 ...], [... α_2 ...] ... >

replace < ... [... $\neg\neg\alpha$ ...] ... > by < ... [... α ...] ... >

Write $((A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C)))$
in CNF.

Proof by resolution

- a *sequence* is a conjunction of lines
- a *line* is a generalized disjunction (clause)
- **growth** of the sequence:
  - if a clause reads as [... β...], insert a new line: [[...$\beta 1$, $\beta 2$ ...]
  - if a clause reads as [... α ...], insert two new lines: [... $\alpha 1$ ...] and [... $\alpha 2$ ...]
  - add new lines by replacing $\neg\neg X$ by X, $\neg$T by $\bot$ and $\neg\bot$ by T
- **resolution**: from lines [... X ...] and [... $\neg X$ ...] create the line [... ... ...], i.e. concatenate the lines leaving aside all occurrences of X and of $\neg X$)
- a *proof of X* by resolution is a sequence including the [$\neg X$] line (*goal*) and containing an empty clause [ ].
- X is a tautology if and only if X has a proof by resolution.

Prove, using the resolution algorithm:

$((A \supset B) \land (B \supset C)) \supset \neg(\neg C \land A)$

a. $[\neg(((A \supset B) \land (B \supset C)) \supset \neg(\neg C \land A))]$

b. $[((A \supset B) \land (B \supset C))]$ développement de a.

c. $[(\neg C \land A)]$ développement de a.

d. $[(A \supset B)]$ développement de b.

e. $[(B \supset C)]$ développement de b.

f. $[\neg A, B)]$ réécriture de d.

g. $[(\neg B, C]$ réécriture de e.

h. $[\neg C]$ développement de c.

i. $[A]$ développement de c.

j. $[B]$ résolvante de f et i.

k. $[C]$ résolvante de g et j.

l. $[\ ]$ résolvante de h et k.

Déduction (syntaxique)

$S \mid\!- X$

X can be proven from S.

$\mid\!- X$

X can be proven.

Show that $S \cup \{X\} \mid\!- Y$ iff $S \mid\!- (X \supset Y)$

Show using deduction theorem that (*modus ponens*)

$$\{P, (P \supset Q)\} \ |- \ Q$$

$\{(P \supset Q)\} \ |- \ (P \supset Q)$ trivial

$\{P, (P \supset Q)\} \ |- \ Q$ (par th. déduction)

Show using deduction theorem that
$((P \supset (Q \supset R)) \supset (Q \supset (P \supset R)))$ is a theorem.

$\{(P \supset (Q \supset R), P, Q\} \ |- R$ (par deux *modus ponens*)

$\{(P \supset (Q \supset R), Q\} \ |- \ (P \supset R)$ (par th. déduction)

$\{(P \supset (Q \supset R)\} \ |- \ (Q \supset (P \supset R))$ (par th. déduction)

$|- \ ((P \supset (Q \supset R) \supset (Q \supset (P \supset R)))$ (par th. déduction)

Axiomatic systems

Un système de Hilbert

schéma d'axiome 1 : $(X \supset (Y \supset X))$

schéma d'axiome 2 : $(X \supset (Y \supset Z)) \supset ((X \supset Y) \supset (X \supset Z))$

schéma d'axiome 3 : $(\bot \supset X)$

schéma d'axiome 4 : $(X \supset \mathrm{T})$

schéma d'axiome 5 : $(\neg\neg X \supset X)$

schéma d'axiome 6 : $(X \supset (\neg X \supset Y))$

schéma d'axiome 7 : $(\alpha \supset \alpha_1)$

schéma d'axiome 8 : $(\alpha \supset \alpha_2)$

schéma d'axiome 9 : $((\beta_1 \supset X) \supset ((\beta_2 \supset X) \supset (\beta \supset X)))$

règle d'inférence (*modus ponens*) : $\dfrac{X \quad (X \supset Y)}{Y}$

Montrer que $((\neg X \supset X) \supset X)$ est un théorème, pour toute formule X.

On part de l'axiome 9 avec $\beta = (\neg X \supset X)$:

$(\neg\neg X \supset X) \supset ((X \supset X) \supset ((\neg X \supset X) \supset X))$

Or $(\neg\neg X \supset X)$ est un axiome (sh. axiome 5).

Si l'on prouve $(X \supset X)$, alors on a le résultat par modus ponens.

Or $(X \supset X)$ résulte de la séquence suivante :

$(X \supset ((X \supset X) \supset X)) \supset ((X \supset (X \supset X)) \supset (X \supset X))$ (sh. axiome 2 et

$X \supset ((X \supset X) \supset X)$ sh. axiome 1 avec $Y = (X \supset X)$)

$(X \supset (X \supset X)) \supset (X \supset X)$ (par modus ponens)

$(X \supset (X \supset X))$ (sh. axiome 1)

$(X \supset X)$ (par modus ponens)

Soundness and completeness

Soundness

F is a propositional formula and *S* is a set of propositional formulas.

If there is a sequence that derives from $S \cup \{\neg F\}$ and that contains the empty clause, then $S \models F$.

Completeness

F is a propositional formula and *S* is a set of propositional formulas.

If $S \models F$, then there is a sequence that derives from $S \cup \{\neg F\}$ and that contains the empty clause.

(time or space) Complexity

- SAT (satisfiability of a propositional expression) is an NP-complete problem.

- SAT restricted to Horn clauses is a P-complete problem.