Predicate Logic

Guy Perrier

Summary

1 - Introduction

- Queries and rules into databases can be expressed by means of **logic**.
- **Propositional logic** cannot go beyond yes/no queries and cannot express general rules.
- Sophisticated queries and general rules need **first order logic**, which deals with **predicates** and **quantification** over arguments of predicates.
- A predicate is a proposition the truth value of which depends on arguments. Some arguments can be **variable** so that they can be **quantified**.
- The creation of the Predicate Logic is due to Gottlob Frege (1879)

1 - Introduction

- Example : genealogy

Jean Muller a comme père Pierre Muller.	pere (pm, jm)
Jean Muller n'est pas une femme.	¬ femme (jm)
Tout parent est une personne.	$\forall$ x $\forall$ y (parent (x,y) ⇒ personne (x))
Tout homme a un père.	$\forall$ x (homme (x) ⇒ $\exists$ y pere (y, x))
Tout homme a un père et une mère.	$\forall$ x (homme (x) ⇒ $\exists$ y pere (y, x)) $\wedge$ $\exists$ z mere (z, x))
Trouver les personnes qui ont Pierre Muller comme père.	{ x | pere(pm, x)}
Trouver les personnes qui n'ont aucun enfant.	{ x | ¬$\exists$ y pere(x,y) $\wedge$ ¬$\exists$ z mere(x, z)}

2 - First order languages : alphabet

- **Logical connectives** : ¬ (not), ∧ (and), ∨ (or), ⇒ (implies), ⇔ (equivalent)
- **Logical constants** : true , false
- **Quantifiers** : ∀ (forall), ∃ (exists)
- **Variables** : a countable set V of variables denoted by x, y , z …
- A **signature** Σ = (Rel , Funct) composed of :
  - a set Rel of **relation symbols** denoted by p, q, r ….

    Every relation symbol is associated with an integer representing the arity of the relation.

    A relation with arity 0 is called an **atomic proposition**.
  - A set Funct of **function symbols** denoted by f, g, h …

    Every function symbol is associated with an integer representing the arity of the function.

    A function with arity 0 is called a **constant**.

2 - First order languages : terms

- The syntax of terms is defined by the following grammar :

$$T \rightarrow v \mid f_0 \mid f_1 (T) \mid f_2 (T, T) \mid f_3 (T, T, T)$$

where v is any element of V and f_n is any element of Funct with the **arity** n.

- The set of variables present in a term t is denoted by Var(t). If Var(t) = Ø, then t is said to be a **closed term**.

- The set of terms built over a signature ∑ is denoted by **Term(∑)**.

2 - First order languages : formulas

- The syntax of formulas is defined by the following grammar :

  $F \to true \mid false \mid p_0 \mid p_1\,(T) \mid p_2\,(T,T) \mid p_3\,(T,T,T)$

  $F \to \neg F \mid F \wedge F \mid F \vee F \mid F \Rightarrow F \mid F \Leftrightarrow F \mid (F) \mid \forall x\, F \mid \exists x\, F$

  $T \to x \mid f_0 \mid f_1\,(T) \mid f_2\,(T, T) \mid f_3\,(T, T, T)$

  where x is any element of V, p_n is any element of Rel with the arity n, f_n is any element of Funct

  and the start symbol is F

- The grammar of formulas is ambiguous. To eliminate ambiguity, a priority is defined between the logical operators :

  - $\neg$ $\forall$ $\exists$
  - $\wedge$
  - $\vee$
  - $\Rightarrow$
  - $\Leftrightarrow$

- The set of formulas defined over a signature Σ is denoted by **Form(Σ)**

2 - First order languages : free and bound variables

- if F is a formula and if x is a variable, the **occurrences of x in F** are **bound** or **free** according to the fact that they are in the scope G of a F subformula in the form ∀ x G or ∃ x G.

- All variables that have a free occurrence at least in a formula F constitute the set FV(F) of the **free variables** of F. A formula that has no free variables is called a **closed formula**.

2 - First order languages : substitution of variables by terms

- if t is a term, the substitution of the variable x by the term t_0 in the term t is a term which is denoted by $t[t_0 /x]$ and which results from replacing all occurrences of x in t with t_0 .

- if F is a formula, the substitution of the variable x by the term t_0 in the formula F is a formula which is denoted by $F[t_0 /x]$ and which results from replacing all **free occurrences** of x in t with t_0 , provided that this entails no **capture of free variable**.

- A capture of a free variable occurs when a variable of the substituted term becomes bound in the substitution. Captures are avoided by **renaming** problematic bound variables.

2 - First order languages

1. Model the following sentences with first order logic formulas :
   a) *Tous les lions sont féroces.*
   b) *Quelques lions ne boivent pas.*
   c) *Aucun singe n'est soldat.*
   d) *Tous les singes sont malicieux.*
   e) *Tous les singes aiment une guenon.*

2. Model the following sentences with first order logic formulas by using the signature given in the course, plus the predicates *frere*, *sœur* and *descendant*:
   a) *Jean Muller est frère d'Annie Muller*
   b) *Jean Muller et Annie Muller ont les mêmes parents*
   c) *Si un individu a le même parent qu'un autre individu et s'il est un homme, alors il est frère du second.*
   d) *Le fait qu'un individu quelconque est descendant d'un autre individu quelconque est équivalent au fait que le second est parent du premier ou parent d'un troisième individu dont le premier est descendant.*

2 - First order languages

3. Give the parse tree of each following formula; for every variable present in these formulas, give their free and bound occurrences; finally, determine if the formulas are closed:

   a) $\forall x \exists y\ pere(x,y) \Rightarrow homme(x)$

   b) $nom(i1) = muller \vee femme(z) \wedge homme(y)$

   c) $\forall x \forall y\ (nom(x) = y \wedge homme(x))$

   d) $\neg\ age(x) = 50 \Rightarrow \exists x\ pere(x,y)$

   e) $\exists x\ (\forall x\ (pere(x, i1) \wedge nom(x) = muller) \Rightarrow femme(x))$

4. Give the parse tree of each following formula; for every variable present in these formulas, give their free and bound occurrences; finally, determine if the formulas are closed:

   a) $(\forall x\ p(x)) \vee (\exists y\ f(x) = y)$

   b) $\forall x \forall y\ (x=y \Rightarrow f(x) = f(y))$

   c) $\exists x\ (\forall x\ p(x, f(a)) \wedge q(x, b)) \Rightarrow r(x)$

2 - First order languages

5. Compute the following substitutions :

   a) $(age(x) = 50 \ \wedge \neg (pere(i1,x)))\ [i2/x]$

   b) $(\forall\ x\ homme(x) \vee (pere(x,y)))[i1/x]$

   c) $(age(x)=y \ \Rightarrow prenom(x) = prenom(z))[i2/x][48/y]$

   d) $(\exists\ x\ (\forall\ x \neg\ pere(x,x)) \wedge homme(x) \Rightarrow pere(x,y))[i3/x]$

6. Compute the following substitutions :

   a) $((\forall\ x\ p(x)) \vee (\exists\ y\ f(x) = y))\ [f(a)/x]$

   b) $(\forall\ x\ \forall\ y\ (x=y \ \Rightarrow f(x,z) = f(y,z)))\ [g(x,y)/z]$

   c) $(\exists\ x\ (\forall\ x\ p(x, f(a)) \wedge q(x, b)) \Rightarrow r(x))\ [f(x)/x]$

3 - First order models : definition

- The goal : to give a **semantics** to the notion of true formula.

- The notion of first order model is due to Alfred Tarski (1933).

- An **interpretation** I of a first order language L is defined as follows:

  ✓ A set D_I considered as the **interpretation domain**,

  ✓ A map of every function symbol f with arity n to a **function** f_I from D^n to D,

  ✓ A map of every predicate symbol p with arity n to a **subset** p_I of D^n, representing the subdomain where the predicate is true.

3 - First order models : interpretation of terms

- For an interpretation I of a language and a valuation Val of its variables, the interpretation $t_{I,Val}$ of any term t is an element of D_I defined recursively as follows:
  - ✓ If t is a variable x, then $t_{I,Val} = Val(x)$,
  - ✓ If $t = f(t_1, \ldots, t_n)$, then $t_{I,Val} = f_I(t_{1I,Val}, \ldots, t_{nI,Val})$.

3 - First order models : interpretation of formulas

- For an interpretation I of a language and a valuation Val of its variables, the interpretation $F_{I,Val}$ of any formula F is a truth-value defined recursively as follows:

  - ✓ If F = true, then $F_{I,Val} = 1$ and if F = false, then $F_{I,Val} = 0$;
  - ✓ If $F = p(t_1, \ldots, t_n)$, then $F_{I,Val} = 1$ if $(t_{1I,Val}, \ldots, t_{nI,Val}) \in p_I$ and $F_{I,Val} = 0$ if $(t_{1I,Val}, \ldots, t_{nI,Val}) \notin p_I$
  - ✓ If F is built with a logical connective, its interpretation stems from the interpretation of its components according to the interpretation of the connective;

3 - First order models : interpretation of formulas

- ✓ Interpretation of logical connectives :

F	¬ F
0	1
1	0

F	G	F ∧ G
0	0	0
0	1	0
1	0	0
1	1	1

F	G	F ∨ G
0	0	0
0	1	1
1	0	1
1	1	1

F	G	F ⇒ G
0	0	1
0	1	1
1	0	0
1	1	1

F	G	F ⇔ G
0	0	1
0	1	0
1	0	0
1	1	1

3 - First order models : interpretation of formulas

*Notation : if **Val** is a valuation, **x** a variable and **a** an element of the domain, then **Val [x:= a]** is a valuation that coincides with **Val** for any variable different from **x** and assigns the value **a** to **x**.*

- ✓ If F = ∀ x G and if for any element *a* of the domain, $G_{I,Val\,[x:=a]} = 1$, then $F_{I,Val} = 1$; otherwise, if there exists some element a of the domain such that $G_{I,Val\,[x:=a]} = 0$, then $F_{I,Val} = 0$.

- ✓ If F = ∃ x G and if for some element *a* of the domain, $G_{I,Val\,[x:=a]} = 1$, then $F_{I,Val} = 1$; otherwise, if for any element a of the domain $G_{I,Val\,[x:=a]} = 0$, then $F_{I,Val} = 0$

3 - First order models

1. Consider a signature $\sum$ = { Funct, Rel} such that: Funct = { min_0 , suc_1 , $plus_2$} and Rel = {$pair_1$, inf_2 , $=_2$ }. Consider the interpretation I with the domain D_I = {1,2,3,4} and the following interpretations for the function and relation symbols:

$min_I = 1$

$suc_I = \{ 1 \longrightarrow 2, 2 \longrightarrow 3, 3 \longrightarrow 4, 4 \longrightarrow 1\}$

$plus_I = \{ (1,1) \longrightarrow 2, (1,2) \longrightarrow 3, (1,3) \longrightarrow 4, (2,2) \longrightarrow 4\}$

$pair_I = \{ 2, 4\}$

$inf_I = \{(1,2), (2,3), (3,4), (4,1)\}$

$=_I \; = \{(1,1), (2,2), (3,3), (4,4)\}$

Give the interpretation of the following formulas with I :

a) pair (min)

b) inf(min, suc(min)) $\vee$ inf(plus(min, min), min)

c) plus(suc(min), min) = suc(suc(min))

d) $\forall$x inf(x, min)

e) $\exists$x $\neg$ (inf(x, min) $\Rightarrow$ (x = min))

3 - First order models

2. In the formulas below, *femme*, *homme* and *frere* are relation names and *a* is constant term. For every following formula, find a model and a counter-model with minimal domains.

a) $femme(a) \wedge \neg\ homme(a)$

b) $\exists\ x\ homme(x) \Rightarrow \forall\ x\ homme(x)$

c) $\forall\ x \forall\ y\ (frere(x,y) \Rightarrow homme(x))$

d) $\forall\ x\ \ \exists\ y\ (frere(x,y) \vee \mathrm{frere}(y,x))$

e) $femme(a) \wedge \forall\ x\ \ \neg\ femme(x)$

3 - First order models

3. Consider the following closed formulas:

- $A = \forall x \forall y \forall z\ (rel(x,y) \wedge rel(y,z) \Rightarrow rel(x,z))$
- $B = \forall x\ \exists y\ rel(y,x)$
- $C = \forall x \forall y\ (rel(x,y) \Rightarrow rel(y,x))$

Determine if the following interpretations are models of these formulas :

a) The set of natural numbers where *rel(x,y)* is interpreted as *“x is strictly less than y”*.

b) The straight lines of the plane where *rel(x,y)* is interpreted as *“ x is perpendicular to y”*.

3 - First order models : interpretation of formulas

- The interpretation of a formula F depends only on I and on the value of Val for the free variables of F. If F is closed, its interpretation depends only on I.

- If the interpretation of a closed formula F in an interpretation I of its language is 1, I is a **model** of F or F **satisfies** I, which is written: $I \models F$

3 - First order models : interpretation of formulas

- If every interpretation of a closed formula F is a model, then F is a **tautology**, which is written : $\models F$
- If a closed formula F has a model, F is **satisfiable**, otherwise F is **inconsistent**.
- If every model of a closed formula F is a model of a closed formula G, G is a **logical consequence** of F, which is written: $F \models G$
- If F and G are mutual logical consequences of themselves, they are **logically equivalent**.

3 - First order models : theories

- A theory is defined from a set of **axioms**, that is a set of formulas given as true.
- The theory, associated with a set of axioms, is the set of all formulas that are logical consequence of a conjunction of axioms.
- A theory is **consistent** if the logical constant *false* does not belong to the theory.
- A theory is **complete** if, for any formula, either the formula or its negation belongs to the theory.

3 - First order models

4. For each formula below, determine if it is a tautology; If not, determine if it is satisfiable or inconsistent.

   a) $\exists x\, p(x) \Rightarrow \forall x\, p(x)$

   b) $\forall x \;\; \exists y \;\; (p(x) \Rightarrow q(y))$

   c) $\forall x \;\; \exists y\, (p(x) \wedge q(y) \vee \; \neg\, q(x) \wedge \neg\, p(y))$

   d) $\exists y\, \forall x\, ((p(x) \vee \; q(y)) \wedge (\neg\, q(x) \vee \neg\, p(y)))$

5. We consider a theory with two axioms expressed in French :

a) *Toute personne qui a un chien est heureuse*

b) *Jean est heureux.*

Prove that the following assertion does not belong to the theory: "*Jean a un chien*"

3 - First order models

6. If F and G are any formulas, show the logical equivalence of:

   a) $\neg \forall x F$ and $\exists x \neg F$

   b) $\forall x F \wedge G$ and $\forall x (F \wedge G)$ if x is not free in G

   c) $\exists x (F \vee G)$ and $\exists x F \vee \exists x G$

   d) $\forall x F \Rightarrow G$ and $\exists x (F \Rightarrow G)$ if x is not free in G

4 - The Predicate Calculus : introduction

- The goal : to give an automatic procedure for determining if a formula is inconsistent, a tautology or a logical consequence of another formula.
- Different frameworks can be used to define **formal systems** complying with this goal : Hilbert systems, sequent calculus, resolution …
- We use a **natural deduction** framework.

4 - The Predicate Calculus : definition

- To express that a formula F is a logical consequence of a set of formulas F_1 , F_2 … F_n , we use a **deduction** F_1 , F_2 … F_n |- F

- A deduction F_1 , F_2 … F_n |- F is valid if it is possible to construct a **proof** of the **conclusion** F from the **hypotheses** F_1 , F_2 … F_n .

4 - The Predicate Calculus : definition

- A proof of the deduction $F_1 , F_2 \ldots F_n \vdash F$ is a **tree** the nodes of which are formulas :

  ✓ The root of the tree is the conclusion F of the proof.

  ✓ The leaves of the tree are the hypotheses $F_1 , F_2 \ldots F_n$ of the proof.

  ✓ Every mother/daughters link in the tree represents an inference wich is justified by an **inference rule**.

4 - The Predicate Calculus : definition

$\neg\, homme(a) \Rightarrow femme(a)$ $[\, \neg\, homme(a)\,]$ ①

-- $\Rightarrow E$

$[\, homme(a)\,]$ ① $femme(a)$

-- $Ex\ M$ -- $\vee\ I$ -- $\vee\ I$

$homme(a) \vee \neg\, homme(a)$ $homme(a) \vee femme(a)$ $homme(a) \vee femme(a)$

-- $\vee\ E$ ①

$homme(a) \vee femme(a)$

4 - The Predicate Calculus : inference rules

- A proof in natural deduction is defined inductively as follows:

✓ **Initialisation**: The single node F is a proof of F |- F.

✓ **Elimination of ⇒** : From a proof tree of Γ |- $F \Rightarrow G$ and another proof tree of Γ |- F, we obtain a proof tree of Γ |- G by adding the new link :

$$\frac{F \Rightarrow G \qquad F}{G}$$

✓ **Introduction of ⇒ :** From a proof tree of Γ, F |- G, we obtain a proof tree of Γ |- $F \Rightarrow G$ by cancelling all leaves F of the initial tree and by adding the new link :

$$\frac{\begin{array}{c}[F]\ ① \\ \vdots \\ G\end{array}}{F \Rightarrow G}\ ①$$

4 - The Predicate Calculus : inference rules

- ✓ **Elimination** of $\wedge$: From a proof tree of Γ |- $F \wedge G$, we obtain a proof tree of Γ |- F, by adding the new link :

$$\frac{F \wedge G}{F}$$

and a proof tree of Γ |- G, by adding the new link :

$$\frac{F \wedge G}{G}$$

- ✓ **Introduction of** $\wedge$ *:* From a proof tree of Γ_1 |- F and another proof tree of Γ_2 |- G, we obtain a proof tree of $\Gamma_1 \cup \Gamma_2$ |- $F \wedge G$ by adding the new link :

$$\frac{F \quad G}{F \wedge G}$$

4 - The Predicate Calculus : inference rules

- ✓ **Elimination of ∨ :** From proof trees of $\Gamma_1 \vdash F \vee G$, $\Gamma_2, F \vdash H$ and $\Gamma_3, G \vdash H$, we obtain a proof tree of $\Gamma_1 \cup \Gamma_2 \cup \Gamma_3 \vdash H$ by cancelling *F* and *G* in the hypotheses of the two last subtrees and by adding the new link :

$$\dfrac{F \vee G \qquad \begin{matrix}[F]\ ① \\ \vdots \\ H\end{matrix} \qquad \begin{matrix}[G]\ ① \\ \vdots \\ H\end{matrix}}{H}\ ①$$

- ✓ **Introduction of ∨ :** we obtain a proof tree of $\Gamma \vdash F \vee G$ from a proof tree of $\Gamma \vdash F$, by adding the new link :

$$\dfrac{F}{F \vee G}$$

and from a proof tree of $\Gamma \vdash G$, by adding the new link : $\dfrac{G}{F \vee G}$

4 - The Predicate Calculus : inference rules

- ✓ **Elimination** of ¬ : From proof trees of Γ_1 |- *F* and Γ_2 |- ¬*F* , we obtain a proof tree of $\Gamma_1 \cup \Gamma_2$ |- *false* , by adding the new link :

$$\frac{F \quad \neg F}{false}$$

- ✓ **Introduction of ¬ :** From a proof tree of Γ, *F* |- *false,* we obtain a proof tree of Γ |- ¬*F* by cancelling all hypotheses *F* and by adding the new link :

[F] ➀

:

:

false

------------ ➀

¬*F*

4 - The Predicate Calculus : inference rules

- ✓ **Elimination of *false* :** From a proof tree of Γ |- *false,* we obtain a proof tree of Γ |- *F* by adding the new link :

$$\frac{\mathit{false}}{F}$$

- ✓ **Excluded middle *:*** |- *F* ∨ ¬*F* is an axiom :

$$\frac{}{F \lor \neg F}$$

4 - The Predicate Calculus : inference rules

- ✓ **Elimination** of ∀ : From a proof tree of Γ |- ∀ *x F*, we obtain a proof tree of Γ |- *F[t/x]* , by adding the new link :

$$\frac{\forall\, x\, F}{F[t/x]}$$

- ✓ **Introduction of ∀ :** From a proof tree of Γ |- *F*, such that the variable *x* is not free in Γ, we obtain a proof tree of Γ |- ∀ *x F* by adding the new link :

$$\frac{F}{\forall\, x}$$

4 - The Predicate Calculus : inference rules

- ✓ **Elimination of ∃ :** From proof trees of Γ_1 |- ∃ *x F* and Γ_2, *F* |- *G*, such that the variable *x* is not free in Γ_2 , *G*, we obtain a proof tree of $\Gamma_1 \cup \Gamma_2$ |- *G* by cancelling all hypotheses *F* in the second subtree and by adding the new link :

[F] ①

⋮

∃ *x F*   *G*

-------------- ①

G

- ✓ **Introduction of ∃ :** From a proof tree of Γ |- *F [t/x]* , we obtain a proof tree of Γ |- ∃ *x F* , by adding the new link :

F [t/x]

∃ *x F*

4 - The Predicate Calculus : soundness and correctness

- **Soundness** : if there is a proof the deduction F_1 , F_2 … F_n |- F , then F is a logical consequence of $F_1 \wedge F_2 \wedge \ldots \wedge F_n$: F_1 , F_2 … F_n |= F .

- **Completeness** (Kurt Gödel, 1933): if F is a logical consequence of $F_1 \wedge F_2 \wedge \ldots \wedge F_n$ (F_1 , F_2 … F_n |= F), then there is a proof the deduction F_1 , F_2 … F_n |- F .

- Corollary : the set of **theorems** (formulas provable without hypotheses) identifies with the set of **tautologies**.

4 - The Predicate Calculus

1. We consider a database constituted of facts and rules :

- The facts are : homme(i1), pere(i2,i1), mere(i3, i1), nom(i1)= muller, prenom(i1)= jean, pere(i4, i2), nom(i2)= muller, prenom(i1)= pierre.
- The rules are : $\forall$ x (femme(x) $\Rightarrow$ $\neg$homme(x))

  $\forall$ x y (pere(x,y) $\Rightarrow$ parent(x,y) $\wedge$ homme(x))

  $\forall$ x y (mere(x,y) $\Rightarrow$ parent(x,y) $\wedge$ femme(x))

  From all facts and rules considered as hypotheses, prove the following formulas :

a) femme(i3)

b) $\neg$ femme(i2)

c) $\exists$ x $\exists$ y (parent(x,y) $\wedge$ nom(x) = muller)

4 - The Predicate Calculus

2. If F and G are any formulas, prove the following theorems:

a) $F \Leftrightarrow \neg\neg F$

b) $\neg F \vee G \Leftrightarrow F \Rightarrow G$

c) $\forall x\, F \wedge G \Leftrightarrow \forall x\, (F \wedge G)$ if x is not free in G

d) $\exists x\, (F \vee G) \Leftrightarrow \exists x\, F \vee \exists x\, G$

4 - The Predicate Calculus

3. Consider a theory with equality. The equality axioms are :

- $\forall\ x\ (x = x)$
- $\forall\ x\ y\ ((x = y) \Rightarrow (y = x))$
- $\forall\ x\ y\ z\ ((x = y) \wedge (y = z) \Rightarrow (x = z))$

Moreover, the theory has two functions, a unary function *suc* and a binary function *plus*, and a constant *0,* which verify the following axioms :

- $\forall\ x\ plus(x, 0) = x$
- $\forall\ x\ y\ plus(x, suc(y)) = suc(plus(x,y))$
- $\forall\ x\ y\ suc(x) = suc(y) \Leftrightarrow x = y$

Prove the following formulas from the axioms of the theory:

a) $plus(0, suc(0)) = plus(suc(0), 0)$

b) $plus(suc(0), suc(0)) = suc(suc(0))$