LA LOGIQUE DES PRÉDICATS ET PROLOG

"On pourrait bien penser tout d'abord que la proposition 7+5 = 12 est une simple proposition analytique qui résulte, suivant le principe de contradiction, du concept d'une somme de 7 et de 5. Mais à y regarder de plus près, on trouve que le concept de la somme de 7 et de 5 ne contient rien de plus que la réunion de deux nombres en un seul, sans que l'on pense aucunement quel est ce nombre unique qui les comprend tous deux. Le concept de 12 n'est nullement pensé par cela seul que je pense simplement cette réunion de 7 et de 5, et j'aurai beau analyser tant que je voudrai le concept d'une somme possible, je n'y rencontrerai cependant pas le nombre 12."

Kant
Critique

La logique propositionnelle

Une *proposition* est une phrase déclarative qui a une valeur logique vraie ou fausse. La logique propositionnelle permet de construire de nouvelles propositions à partir de propositions de bases appelées propositions atomiques et de cinq *connecteurs logiques* "et" (∧), "ou" (∨), "non" (¬), "si" (⇒) et "si et seulement si" (⇔).

P1 ≡ "il fait beau"

P2 ≡ "il ne pleut pas"

P3 = P1∧P2 = "il fait beau" et "il ne pleut pas"

Donnons une définition plus formelle de la *logique propositionnelle*:

Un atome est une proposition.
Si P est une proposition alors ¬P est une proposition.
Si P1 et P2 sont des propositions alors P1∧P2, P1∨P2, P1⇒P2 et P1⇔P2 sont des propositions.
Toute construction faite par application des règles précédentes un nombre fini de fois est une proposition.
Tout autre mode de construction ne conduit pas à une proposition.

L'interprétation d'une proposition
L'interprétation (ou *évaluation*) en logique propositionnelle est une application I de l'ensemble des propositions sur l'ensemble des valeurs de vérité (ou *valuations*) {Vraie, Fausse}.

Notation: V = Vraie et F = Fausse.

P	Q	P∧Q	P∨Q	¬P	P⇒Q	P⇔Q
F	F	F	F	V	V	V
F	V	F	V	V	V	F
V	F	F	V	F	F	F
V	V	V	V	F	V	V

Une proposition est une *tautologie* (ou proposition valide) si sa valeur de vérité est Vraie pour toute interprétation.

$$P \vee \neg P$$

Une proposition est *contradictoire* (ou inconsistante) si sa valeur de vérité est Fausse pour toute interprétation.

$$P \wedge \neg P$$

Deux propositions sont dites logiquement *équivalentes* (ou égales) si leurs valeurs de vérité sont égales pour toute interprétation.

$P \vee Vraie$	$= Vraie$
$P \wedge Vraie$	$= P$
$P \vee Fausse$	$= P$
$P \wedge Fausse$	$= Fausse$
$P \wedge (\neg P)$	$= Fausse$
$P \vee (\neg P)$	$= Vraie$
$\neg (\neg P)$	$= P$
$P \vee Q$	$= Q \vee P$
$P \wedge Q$	$= Q \wedge P$
$\neg (P \wedge Q)$	$= (\neg P) \vee (\neg Q)$
$\neg (P \vee Q)$	$= (\neg P) \wedge (\neg Q)$
$(P \vee Q) \vee R$	$= P \vee (Q \vee R)$
$(P \vee Q) \wedge R$	$= (P \wedge R) \vee (Q \wedge R)$
$(P \wedge Q) \wedge R$	$= P \wedge (Q \wedge R)$
$(P \wedge Q) \vee R$	$= (P \vee R) \wedge (Q \vee R)$
$P \Rightarrow Q$	$= (\neg P) \vee Q$
$P \Leftrightarrow Q$	$= (P \Rightarrow Q) \wedge (Q \Rightarrow P)$

Une proposition est dite *sous forme normale conjonctive* si elle est une conjonction de disjonctions de propositions ou de négations de propositions, ex:

$$P \wedge Q \wedge (\neg R) \wedge (\neg P \vee S)$$

Une proposition est dite *sous forme normale disjonctive* si elle est une disjonction de conjonctions de propositions ou de négations de propositions, ex:

$$P \vee Q \vee (\neg R) \vee (\neg P \wedge S)$$

Il est aisé de démontrer que toute proposition admet deux propositions logiquement équivalentes sous forme normale conjonctive et disjonctive.

> Pour passer d'une proposition quelconque à une forme normale il suffit d'éliminer les $\Rightarrow$ et les $\Leftrightarrow$, de ramener les $\neg$ près des atomes et d'utiliser la distributivité pour mettre la proposition sous l'une des deux formes normales.

Une proposition Q est une *conséquence logique* des propositions P1...Pn

> si et seulement si $(P1 \wedge ... \wedge Pn) \Rightarrow Q$ est une tautologie
> ce qui est équivalent à dire que $(P1 \wedge ... \wedge Pn \wedge \neg Q)$ est une contradiction.

P1...Pn sont appelés les *axiomes*, *postulats* ou *prémisses* de la proposition Q, qui est alors considérée comme un *but*.

Q est une conséquence logique de (P⇒Q) et de P.

Cette règle est appelée le *Modus ponens*

"il pleut"; si "il pleut" alors "il ne fait pas beau";
par le modus ponens on en déduit qu'"il ne fait pas beau".

Considérons les deux propositions:

"Socrate est un homme"
"Tout homme est mortel"

Avec la logique propositionnelle il n'est pas possible de construire la proposition "Socrate est mortel" comme conséquence des deux propositions précédentes; il faut une logique plus fine: la logique des prédicats.

La logique des prédicats

La logique des prédicats aussi appelée logique du premier ordre est fondée sur les notions de *terme*, *prédicat* et *formule*.

L'ensemble des symboles de la logique du premier ordre peut se subdiviser en trois sous-ensembles:

sous-ensemble de variables x,y,z,...
sous-ensemble de fonctions f,g,h,...
sous-ensemble de prédicats P,Q,R,...

A chaque symbole de fonction et de prédicat est associé un nombre entier positif ou nul: son arité.

La définition récursive d'un terme est:

une variable est un terme.
f(a1,...,an) est un terme si et seulement si f est un symbole de fonction d'arité n, et où les arguments ai sont des termes.

P(p1,...,pn) est un prédicat ou atome si et seulement si P est un symbole de prédicat d'arité n, et où les paramètres pi sont des termes.

La logique des prédicats est fondée sur les formules comme la logique propositionnelle est fondée sur les propositions.

La définition récursive d'une formule (ou expression du premier ordre) est:

un atome est une formule.

si F est une formule, alors ¬F est une formule.

si F1 et F2 sont des formules, alors F1∧F2, F1∨F2, F1⇒F2 et F1⇔F2 sont des formules.

si F est une formule et x une variable libre dans F, alors ∀x F et ∃x F sont des formules.

Toute construction faite par application des règles précédentes un nombre fini de fois est une formule.

Tout autre mode de construction ne conduit pas à une formule.

Deux notions apparaissent: les *variables* et les *quantificateurs*

le *quantificateur existentiel*

∃(∃x F peut se lire "il existe un x tel que F")

le *quantificateur universel*

∀(∀x F peut se lire "pour tout x on a F")

La *portée d'un quantificateur* est l'ensemble des variables atteintes par ce quantificateur.

Une occurrence d'une *variable* dans une formule est dite *liée* si et seulement si cette occurrence se trouve dans la portée du quantificateur qui l'utilise. Dans le cas contraire elle sera dite *libre*.

Une formule sera dite close si et seulement si toutes les occurrences de variables sont liées.

L'interprétation des formules

Une formule s'interprète dans un ensemble non vide appelé domaine (ou environnement).

En ce qui concerne les connecteurs, l'interprétation se fait de la même manière qu'en logique propositionnelle, et pour les formules contenant des quantificateurs:

∃x F est Vraie si et seulement si F est Vraie pour au moins un élément du domaine, et Fausse sinon.

∀x F est Vraie si et seulement si F est Vraie pour tout élément du domaine, et Fausse sinon.

Une formule sera dite contradictoire, valide ou équivalente à d'autres formules exactement avec les même définitions que pour les propositions.

```
(Qx) F(x)∧G          = (Qx)(F(x)∧G)
(Qx) F(x)∨G          = (Qx)(F(x)∨G)
(∀x)F(x)∧(∀x)G(x)    = (∀x)(F(x)∧G(x))
(∃x)F(x)∨(∃x)G(x)    = (∃x)(F(x)∨G(x))
(Q1x)F(x)∧(Q2y)G(y) = (Q1x)(Q2y)(F(x)∧G(y))
(Q1x)F(x)∨(Q2y)G(y) = (Q1x)(Q2y)(F(x)∨G(y))
¬(∀x F(x))           = ∃x ¬F(x)
¬(∃x F(x))           = ∀x ¬F(x)
```

Une formule F est dite *sous forme normale de prénexe* si et seulement si elle est une formule sans quantificateur ou est écrite sous la forme:

(Qx)F'

où Q est un quantificateur, x une variable et F' une formule sous forme normale de prénexe.

Comme pour les formes normales conjonctives et disjonctives de la logique d'ordre zéro, toute formule est équivalente à au moins une formule sous forme normale de prénexe.

1 suppression des ⇒ et ⇔

2 ramener les ¬ près des atomes.

3 renommer les variables liées si nécessaire.

4 utiliser l'équivalence de formule pour déplacer les quantificateurs.

Les clauses

Une formule F est dite sous *forme clausale* si et seulement si elle ne contient pas de quantificateur et est sous forme normale conjonctive ou si elle est de la forme:

$$F = \forall x\ F'$$

où F' est une formule sous forme clausale.

Nous allons introduire maintenant les fonctions de skolémisation qui vont servir à transformer toute formule sous forme normale de prénexe en formule sous forme clausale, c'est-à-dire à éliminer les quantificateurs existentiels.

Considérons une formule sous forme de prénexe:

$$\forall x1,...,\forall x_n,\ \exists y\ F(y,x1,...,xn)$$

et la formule:

$$\forall x1,...,\forall xn \quad F(fsko(x1,...,xn),x1,...,xn)$$

où $fsko$ est un symbole de fonction.

On démontre que la première formule est inconsistante si et seulement si la deuxième est aussi inconsistante.

De ce fait il est possible de trouver une "bonne" fonction de skolémisation de manière à supprimer les quantificateurs existentiels d'une formule sous forme normale de prénexe et donc la transformer en forme clausale.

Voici la méthode pour passer d'une formule quelconque à une forme clausale:

1 transformer l'expression sous forme normale de prénexe.

2 mettre la partie droite de la formule sans quantificateur sous forme conjonctive.

2.1 si dans la partie de la formule, qui ne contient que des quantificateurs, un ∃ n'est pas précédé par des ∀ remplacer toutes les occurrences de la variable quantifiée par une constante (c'est-à-dire instancier la variable) et supprimer le quantificateur ∃.
2.2 si dans cette même partie le ∃ est précédé de n ∀, remplacer chaque occurrence de la variable quantifiée par une nouvelle fonction à n arguments qui seront les variables quantifiées par les n ∀ et supprimer le quantificateur ∃.

Exemple:

$(\forall x)K(x) \Rightarrow ((\exists x)F(x)\vee(\exists x)G(x))\wedge((\forall y)H(y)\vee(\exists z)I(z))$
=
$(\forall x)K(x) \Rightarrow ((\exists x)(F(x)\vee G(x)))\wedge((\forall y)(\exists z)(H(y)\vee I(z)))$
=
$(\forall x)K(x) \Rightarrow (\exists x)(\forall y)((F(x)\vee G(x))\wedge(\exists z)(h(y)\vee I(z)))$
=
$(\forall x)K(x) \Rightarrow (\exists x)(\forall y)(\exists z)((F(x)\vee G(x))\wedge(H(y)\vee I(z)))$
=
$(\neg((\forall x)K(x)))\vee((\exists x)(\forall y)(\exists z)((F(x)\vee G(x))\wedge(H(y)\vee I(z))))$
=
$((\exists x)(\neg K(x)))\vee((\exists x)(\forall y)(\exists z)((F(x)\vee G(x))\wedge(H(y)\vee I(z))))$
=
$(\exists x)((\neg K(x))\vee((\forall y)(\exists z)((F(x)\vee G(x))\wedge(H(y)\vee I(z)))))$
=
$(\exists x)(\forall y)(\exists z)((\neg K(x))\vee((F(x)\vee G(x))\wedge(H(y)\vee I(z))))$
=
$(\exists x)(\forall y)(\exists z)((\neg K(x)\vee F(x)\vee G(x))\wedge(\neg K(x)\vee H(y)\vee I(z)))$

cette formule est sous forme normale de prénexe, introduisons maintenant les fonctions de skolémisation:

on remplace x par une constante: a, et on introduit une fonction de skolémisation: fsko d'arité 1

$(\forall y)(((\neg K(a))\vee G(a)\vee F(a))\wedge((\neg K(a))\vee H(y)\vee I(fsko(y))))$

cette formule est sous forme clausale.

Un litéral est une formule atomique (*litéral positif*) ou la négation d'une formule atomique (*litéral négatif*).

On appelle clause une disjonction de litéraux.

Un ensemble de clauses E correspond à une formule conjonctive $C1\wedge...Cn$ où les Ci sont des clauses; cet ensemble sera noté: E={C1,...,Cn}

formule sous forme clausale:

$(\forall y)((F(a)\vee G(a)\vee(\neg K(a)))\wedge((\neg K(a))\vee H(y)\vee I(fsko(y))))$

on dérive l'ensemble de clauses:

E={(F(a)∨G(a)∨(¬K(a))),((¬K(a))∨H(y)∨I(fsko(y)))}

où y est une variable libre et a une constante.

La notion d'ensemble de clauses est à manipuler avec précaution car à un ensemble de clause correspond une formule, il faut voir cette notion comme une facilité d'écriture.

Le principe de résolution

Une substitution est une application de l'ensemble des variables sur l'ensemble des termes.

Une substitution peut être vue comme un ensemble fini de couples notés vi/ti où vi est une variable et ti est un terme.

S={ x/f(x,a), y/P(a,b,x), z/c }
S(Q(x,y,z,a)) = Q(f(x,a),P(a,b,x),c,a)

Deux substitutions peuvent être composées pour en former une troisième. Cette loi de composition confère à l'ensemble des substitutions une structure de groupe commutatif dont l'élément neutre est la substitution vide.

Deux litéraux P et Q sont dits unifiables s'il existe une substitution telle que les litéraux P et Q transformés par cette substitution sont équivalents.

On dit qu'on unifie un ensemble de clauses E={C_1,...,C_n} si on peut trouver une substitution S qui rende équivalente les clauses C_i transformées par substitutions.

Un ensemble de clauses E implique logiquement une clause C si et seulement si toute interprétation qui satisfait E satisfait C, ce qui est équivalent à dire que E $\cup$ {$\neg$C} est inconsistant. L'implication logique est en logique des prédicats ce qu'est la conséquence logique en logique propositionnelle.

Pour démontrer une implication logique on utilise le principe de résolution qui consiste à partir de deux clauses: L$\vee$ L1 $\vee$...Ln et $\neg$L$\vee$ P1 $\vee$...Pm à déduire la clause résolvante: L1 $\vee$...Ln $\vee$ P1 $\vee$...Pm

La stratégie de démonstration par effacement

Cette stratégie, aussi appelée SL-résolution, sert à montrer si un ensemble de clauses E implique logiquement un litéral L.

Ce qui revient à prouver l'inconsistance de l'ensemble
E'=E $\cup$ {$\neg$L}. Pour ce faire on cherche dans E une clause contenant le litéral L: L$\vee$ L1 $\vee$...Ln, par le principe de résolution on déduit la résolvante L1 $\vee$...Ln. On dit que l'on a effacé $\neg$L de l'ensemble de clauses E'. La stratégie consiste à effacer chaque litéral li jusqu'au moment où l'ensemble de clauses devient vide; or un ensemble de clauses vide est inconsistant. Donc si on arrive à l'ensemble vide cela prouve que le litéral L est vrai.

Effacer une clause revient à effacer tous les litéraux constituant la clause.

Remarque:

Quand on cherche dans l'ensemble des clauses le litéral complémentaire de celui qu'on essaie d'effacer, il est possible de trouver plusieurs clauses contenant ce litéral complémentaire, nous sommes alors en position de choix. Nous en choisissons une, et les autres clauses appelées choix en attente devront être utilisés en cas d'échec avec le choix précédent et ce jusqu'au moment où il n'y aura plus de choix. Ce mécanisme de retour en arrière dans la stratégie d'effacement est appelé remontée ou *backtracking*.

Le fait qu'il existe des points de choix montre le *non-déterminisme* de la stratégie d'effacement; ce problème de non-déterminisme a d'ailleurs été considéré comme le premier problème NP-Complet (Non déterministe Polynomial problème de base) par Cook en 1971.

Le langage PROLOG

Comme son nom l'indique le langage *Prolog* est un langage de programmation en logique, plus précisément en logique des prédicats. Sa date de naissance se situe aux alentours de 1972 et ses parents sont Alain Colmerauer et son équipe de l'université de Marseille.

En Prolog une clause $L_1 \wedge ... \wedge L_n \Rightarrow L$ s'exprime sous une forme particulière L :- L_1 ,..., L_n. dite *clause de Horn* où L s'appelle la tête de clause et L_1 ,..., L_n le corps de la clause.

Une clause avec un corps vide s'appelle un fait.

Une clause ayant un corps non vide s'appelle aussi une règle.

Le langage Prolog est constitué de deux parties: une base de données qui est un ensemble de clauses de Horn (le programme) et un moteur d'inférence dont le mécanisme repose sur la stratégie d'effacement, avec un parcours linéaire de la base de donnée pour la recherche d'unification (la recherche des litéraux servant à l'effacement).

Exécuter un programme Prolog revient à demander au démonstrateur (le *moteur d'inférences*) de prouver un but (un litéral) en utilisant la base de données (faits + règles).

Le mécanisme de backtracking assure lors d'un échec (unification impossible pour trouver une résolvante) au cours de la démonstration d'un sous-but, la reprise de la résolution à partir d'un choix en attente.

Il est possible de forcer ce mécanisme en utilisant explicitement le prédicat "fail" qui est par nature non unifiable.

Il est aussi possible d'empêcher ce mécanisme en utilisant explicitement le prédicat "!" qui se lit "cut".

> Le *prédicat cut* (!) permet de transformer une recherche non déterministe en recherche déterminisme et donc de simuler le comportement d'un choix conditionné (If Then Else).