Introduction aux techniques de l'intelligence artificielle

Alain MILLE
CPE - Lyon
Techniques de l'intelligence artificielle

10 janvier 1999

Table des matières

1 Préambule

Cette introduction a pour but de rappeler les bases de la logique qui sous-tend les mécanismes fondamentaux de l'Intelligence Artificielle, comme déduction, abduction, etc....

2 Calcul des propositions

Une *proposition* est un énoncé déclaratif (ou assertion) possédant par définition une seule *valeur de vérité* : V(vraie) ou F(fausse).

En langage naturel une proposition est en général une affirmation comme :

- la pluie mouille,
- il pleut,
- je suis dehors,
- je suis mouillé.

Ces affirmations sont des éléments d'information atomiques.

Les propositions sont représentées symboliquement par les éléments d'un ensemble dénombrable (un *alphabet*) appelés *littéraux*.

Une application *val* (valeur de vérité) de l'ensemble des propositions $\{X\}$ est définie sur l'ensemble $\{0,1\}$:

$$\begin{aligned} \{X\} &\xrightarrow{val} \{0,1\} \\ val(p) &\longrightarrow \left\{ \begin{array}{c} 0 \\ 1 \end{array} \right\} \end{aligned} \tag{1}$$

2.1 Foncteurs de vérité unaires

Il s'agit d'applications de l'ensemble $\{X\}$ des propositions possibles sur lui-même.

$$\begin{aligned} \{X\} &\xrightarrow{f} \{X\} \\ f(p) &\longrightarrow q \end{aligned} \tag{2}$$

A chacune de ces applications f correspond une application de $\{0,1\}$ sur lui-même :

$$\begin{array}{ccc} p & \xrightarrow{f} & q \\ \Big\downarrow val & & \Big\downarrow val \\ \{0,1\} & \xrightarrow{\varphi} & \{0,1\} \end{array} \tag{3}$$

On a donc :

$$val\,(f\,(p)) = \varphi\,(val\,(p)) \tag{4}$$

Dans le cas de la logique bivalente 4 foncteurs unaires f sont possibles :

p	$val(p)$	$f_1(p)$	$f_2(p)$	$f_3(p)$	$f_4(p)$
V	1	1	1	0	0
F	0	1	0	1	0

FIG. 1 – *Table des foncteurs unaires en logique bivalente*

Dans le reste du texte, la relation d'équivalence $\equiv$ sera remplacée par sa version "opérationnelle" $\leftrightarrow$.
Les foncteurs f_1 et f_4 sontles constantes 1 et 2 respectivement. Le foncteur f_3 est appelé *négation de p*, il est dénoté par $\neg p$. f_2 est la *négation* de f_3 qui possède la même table de vérité que *val*. Cette double négation définit la loi d'**idempotence** :

$$\neg (\neg p) \equiv p \tag{5}$$

2.2 Foncteurs de vérité binaires (interpropositionnels)

Il s'agit d'applications f de l'ensemble produit $X \times X$ dans X, avec X l'ensemble des propositions possibles.

$$X \times X \xrightarrow{f} X \tag{6}$$
$$f(p,q) = r \tag{7}$$

Un tableau similaire à celui présenté en figure 1 peut être établi à partir du diagramme suivant :

$$\begin{array}{ccc} p \times q & \xrightarrow{f} & r \\ {\scriptstyle val \times val}\downarrow & & \downarrow{\scriptstyle val} \\ \{0,1\} \times \{0,1\} & \xrightarrow{\varphi} & \{0,1\} \end{array} \tag{8}$$

avec :

$$val\left(f\left(p,q\right)\right) = \varphi\left(val\left(p\right), val\left(q\right)\right) \tag{9}$$

Ce tableau (figure 2) met en évidence les 16 foncteurs de vérité binaire. Un semantique classique est précisée pour chacun des foncteurs.

					notation	sémantique
p	1	1	0	0		
q	1	0	1	0		
1	1	1	1	1		tautologie
2	1	1	1	0	$\vee$	ou inclusif
3	1	1	0	1	$\Leftarrow$	p si q
4	1	1	0	0		$f_4(p.q) = p$
5	1	0	1	1	$\Rightarrow$	p implique q
6	1	0	1	0		$f_6(p.q) = q$
7	1	0	0	1	$\Leftrightarrow$	p si et seulement si q
8	1	0	0	0	$\wedge$	p et q
9	0	1	1	1	$\mid$	NAND incompatibilité
10	0	1	1	0	$\oplus$	XOR alternative, ou exclusif
11	0	1	0	1		
12	0	1	0	0		
13	0	0	1	1		
14	0	0	1	0		
15	0	0	0	1	$\downarrow$	NOR - rejet
16	0	0	0	0		antilogie

FIG. 2 – *Table de vérité (foncteurs binaires)* $f(p,q) = r$

Les notations sont indicatives et varient selon les auteurs.
La sémantique indiquée doit être considérée avec précaution. Le sens usuel des termes "implique", tautologie, antilogie, etc. sont souvent différents de la stricte interprétation en logique. Ainsi la proposition $[p \Rightarrow q]$ est toujours vraie sauf si $p = 1$ et $q = 0$, ce qui signifie seulement que le faux (0) ne peut impliquer le vrai (0).

2.3 Expressions Bien Formées (EBF)

Les foncteurs binaires permettent de transformer des propositions simples en propositions complexes que l'on appelle le plus souvent des formules. Une formule sera libéllée en majuscules. Toutes les formules possibles (combinant des littéraux de propositions simples, des opérateurs, des connecteurs, des parenthèses) ne sont pas forcément bien formées pour constituer à leur tour une expression trouvant sa valeur dans le système formel. Quatre règles doivent être respectées pour qu'une formule soit bien formée :

- *règle 1 : un littéral de proposition simple est une EBF de même que les constantes logiques V (vrai) et F (faux),*
- *règle 2 : P étant une formule, si P est une EBF alors* $\neg P$ *est une EBF,*
- *règle 3 : P et Q étant deux formules, si P et Q sont des EBF alors* $P \Rightarrow Q$ *est une EBF.*

– *règle 4 : rien n'est une EBF sinon les suites de symboles produites aux règles 1 à 3 précédentes.*

Ces règles permettent de distinguer :

– la fausseté d'une formule qui caractérise un énoncé ayant pour valeur de vérité F,

– l'absence de signification qui caractérise un énoncé dénué de sens,

Par exemple, $[\neg (\Rightarrow pq)]$ n'est pas un énoncé faux mais un énoncé dénué de sens (n'est pas une EBF).

2.4 Calcul des propositions : évaluation des EBF

Le calcul des propositions consiste à trouver la valeur de vérité d'une formule P composée de propositions atomiques $p_1 \dots p_i \dots p_n$.

Deux méthodes sont disponibles :

– la méthode des tables de vérité,

– la méthode de parcours d'arbre de solutions.

Si on considère qu'une tautologie est une formule dont les valeurs de vérité sont toutes vraies, alors une tautologie peut s'appeler une formule valide (on dit aussi qu'une formule valide constitue une loi logique).

La logique *formelle* peut alors être considérée comme la recherche de toutes les tautologies possibles.

Le chapître suivant présente un certain nombre de lois logiques constituant la base de la logique formelle.

2.5 Principales lois logiques (tautologies)

$p \Leftrightarrow p$	loi d'identité
$p \Rightarrow p$	tautologie élémentaire
$p \Leftrightarrow (p \vee p)$ $p \Leftrightarrow (p \wedge p)$	lois d'idempotence des connecteurs $\vee$ et $\wedge$
$\neg (p \wedge \neg p) \Leftrightarrow V$	loi de non contradiction deux formules p et $\neg p$ ne peuvent être toutes les deux vraies.
$p \vee \neg p \Leftrightarrow V$	loi du tiers exclu deux formules p et $\neg p$ ne peuvent toutes les deux fausses.
$\neg\neg p \Leftrightarrow p$	loi de la double négation

FIG. 3 – *Lois élémentaires sur une proposition unique*

Les lois présentées dans la figure 3 sont à la base de nombreuses simplifications. L'absorption est une autre façon de simplifier les formules. Les différentes lois permettant d'autres simplifications (ou développements) avec les connecteurs $\vee$ et $\wedge$ sont présentées dans le tableau suivant (figure 4) :

$(p \vee q) \Leftrightarrow (q \vee p)$ $(p \wedge q) \Leftrightarrow (q \wedge p)$	Commmutativité
$((p \vee q) \vee r) \Leftrightarrow (p \vee (q \vee r))$ $((p \wedge q) \wedge r) \Leftrightarrow (p \wedge (q \wedge r))$	Associativité
$(p \vee (q \wedge r)) \Leftrightarrow (p \vee q) \wedge (p \vee r)$ $(p \wedge (q \vee r)) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	Double distributivité
$p \wedge (p \vee q) \Leftrightarrow p$ $p \vee (p \wedge q) \Leftrightarrow p$	Absorption

FIG. 4 – *Lois de commutativité, associativité, double distributivité et absorption*

Un certain nombre de lois permettent d'établir des dualités d'écriture de connecteurs en termes d'autres connecteurs. Ces lois sont connues sous le nom de lois de **DE MORGAN** (tableau de la figure 5).

$\neg (p \wedge q) \Leftrightarrow \neg p \vee \neg q$ $\neg (p \vee q) \Leftrightarrow \neg p \wedge \neg q$ $(p \Rightarrow q) \Leftrightarrow (\neg p \vee q)$ $(p \Rightarrow q) \Leftrightarrow \neg (p \vee \neg q)$ $(p \Leftrightarrow q) \Leftrightarrow (p \Rightarrow q) \wedge (q \Rightarrow p)$ $(p \vee q) \Leftrightarrow (\neg p \Rightarrow q)$ $(p \wedge q) \Leftrightarrow \neg (p \Rightarrow \neg q)$	Lois de dualité ou lois DE MORGAN

FIG. 5 – *Lois de DE MORGAN*

Le tableau suivant (figure 6) présente quelques lois dites de l'implication.

$(p \Rightarrow q \Leftrightarrow \neg q) \Rightarrow p$	Loi de contraposition
$((p \Rightarrow q) \wedge (q \Rightarrow r)) \Rightarrow (p \Rightarrow r)$	Loi de la transitivité et de l'implication
$((p \Rightarrow q) \wedge (p \Rightarrow \neg q)) \Rightarrow \neg p$	Loi de réduction à l'absurde
$(\neg p \Rightarrow p) \Rightarrow p$	Loi de Clavius
$(p \Rightarrow (q \Rightarrow p)) \Rightarrow p$	Loi de Pierce
$p \Rightarrow (q \Rightarrow p)$	*Verum sequitur ad quodlibet*, une proposition vraie est impliquée par n'importe quelle autre proposition.
$\neg p \Rightarrow (p \Rightarrow q)$	Ex falso sequitur quodlibet (à partir du faux on peut déduire n'importe quoi)
$((p \Rightarrow q) \wedge p) \Rightarrow q$	Modus ponens *(latin pono : poser ou affirmer)*
$((p \Rightarrow \neg q) \wedge q) \Rightarrow \neg p$	Modus tollens *(latin tollo : enlever ou nier)*
$(p \Rightarrow (q \Rightarrow r)) \Leftrightarrow ((p \wedge q) \Rightarrow r)$	Principe "*d'import-export*"

FIG. 6 – *Lois d'implication*

2.6 Substitution et remplacement

L'opérateur de substitution permet de formaliser la propagation d'une formule par le remplacement d'un littéral partout où il apparait dans une autre formule.
L'expression de la substitution s'écrit :

$S(Q, p, P) \qquad (10)$	Substitution de Q à p dans P

Par exemple :

$$\begin{array}{l} P : p \Rightarrow (q \Rightarrow p) \\ Q : p \Rightarrow q \\ S(Q, q, P) = \text{ Substituer } Q \text{ à toutes les occurences de } q \text{ dans } P \\ \text{On obtient } p \Rightarrow ((p \Rightarrow q) \Rightarrow p) \end{array} \qquad (11)$$

Théorème 2.1 *Si P est une tautologie et Q une EBF alors $S(Q, q, P)$ est une tautologie.*

Corollaire du théorème 2.1 *Si R est une tautologie et P une formule constituante de R et si $(P \Rightarrow Q)$ est une tautologie, alors l'expression obtenue en remplaçant P par Q dans une ou plusieurs occurences quelconques de P dans R est une tautologie.*

2.7 La règle d'élimination

C'est une tautologie permettant de simplifier grand nombre de formules. Cette règle (que le lecteur démontrera facilement) indique :

$$(P \vee Q) \wedge (\neg P \vee R) \Rightarrow Q \vee R \qquad (P \text{ est "dissous"} !) \qquad (12)$$

2.8 Jeu de connecteurs fonctionnellement complet

Les 16 connecteurs de la table de vérité décrite en figure 2 peuvent être tous représentés à l'aide des seuls connecteurs :

- conjonction $\vee$
- disjonction $\wedge$
- implication $\Rightarrow$

Ainsi, $p \Rightarrow q$ équivaut à $\neg \vee q$.
On appelle fonctionnellement complet le jeu de connecteurs qui permet d'obtenir tous les autres par définition.
En conséquence toute EBF peut être ramenée à une formule où ne figurent que les opérateurs $\neg$, $\wedge$, $\vee$, formant le concept des *formes normales d'une EBF*.

2.9 Formes normales d'une EBF

Définition 2.1 *On appelle forme normale disjonctive distinguée (FNDD) d'une expression P une forme disjonctive équivalente à P (mêmes valeurs de vérité) et dont tous les membres disjonctifs contiennent* **une et une seule fois chacun** *des littéraux propositionnels figurant dans P.*

Pour écrire la FNDD d'une expression donnée, il suffira d'énumérer chacun des cas où l'expression prend la valeur V (vraie).

Définition 2.2 *On appelle forme normale conjonctive distinguée (FNCD) d'une expression P une forme conjonctive équivalente à P dont tous les facteurs contiennent une et une seule fois chacun des littéraux propositionnels figurant dans P, ces littéraux étant ou non affectés de négation.*

Pour écrire la FNCD d'une expression, il suffira d'inventorier chacun des cas où l'expression P prend la valeur F (faux).

On notera qu'une FNCD est une conjonction de disjonctions élémentaires.
Remarque : Les disjonctions citées dans la définition sont appellées **clauses** et la FNCD est une **forme clausale**.

Ces définitions servent à la résolution de calcul de propositions pour la démonstration automatique de tautologies.

2.10 Forme clausale de KOLAWSKI

La forme clausale de Kowalski est une forme abrégée de la notation clausale précédente :

$$P_1, P_2, \ldots, P_m \Leftarrow Q_1, Q_2, \ldots, Q_n \tag{13}$$

Les virgules à **gauche** sont à lire **ou**,
Les virgules à **droite** sont à lire **et**,
La flêche orientée à gauche se lit **si**,
Les P_i et Q_j sont appelés **clauses**,
Les clauses P_i sont appelées **conclusions**,
Les clauses Q_i sont appelées **conditions**.
La forme généralisée introduit les valeurs F (faux) et V :

$$P_1, P_2, \ldots, P_m, F \Leftarrow Q_1, Q_2, \ldots, Q_n, V \tag{14}$$

Plusieurs cas particuliers sont distingués :

$m > 1$	les conclusions sont multiples, c'est le cas non déterministe.
$m \leq 1$	Cas particulier des clauses dites **clauses de HORN** du nom d'un chercheur américain (1951)
$m = 1$ $n > 0$	$P \Leftarrow Q_1, Q_2, \ldots, Q_n$ clauses définies. Il existe une conclusion. C'est le cas déterministe.
$m = 1$ $n = 0$	$P \Leftarrow V$ clause définie inconditionnelle : c'est un **fait** ou une **assertion** ou encore un **axiome**.
$m = 0$ $n > 0$	$F \Leftarrow Q_1, Q_2, \ldots, Q_n$ négation formelle de la conjonction des Q_j
$m = 0$ $n = 0$	$F \Leftarrow V$ une antilogie fondamentale

2.11 La preuve par l'absurde ou réfutation

Prouver qu'une clause est vraie revient à prouver que la négation de cette clause est une contradiction (antilogie).

Ce type de preuve est bien traité dans le cas des clauses de HORN. La résolution est basée sur trois *schémas* appliqués de manière systèmatique.

Schéma 1

$$\begin{array}{ll} P \Leftarrow Q, R & P \text{ si } Q \text{ et } R \\ & \text{et} \\ F \Leftarrow P & P \text{ est } faux \\ \hline & \text{alors} \\ F \Leftarrow Q, R & (Q \text{ et } R) \text{ est faux, c'est à dire non vrais en même temps} \end{array}$$

Schéma 2

$$\begin{array}{ll} P \Leftarrow V & P \text{ est vrai} \\ & \text{et} \\ F \Leftarrow P, Q & (P \text{ et } Q) \text{ est faux (pas vrais en même temps)} \\ & \text{alors} \\ \hline F \Leftarrow Q & Q \text{ est faux} \end{array}$$

Schéma 3

$$\begin{array}{ll} P \Leftarrow V & P \text{ est vrai} \\ & \text{et} \\ F \Leftarrow P & P \text{ est faux} \\ & \text{alors} \\ \hline F \Leftarrow V & \textbf{contradiction} \end{array}$$

Schéma 4 En généralisant :

$$\begin{array}{l} P_1 \Leftarrow Q_1, Q_2, \ldots, Q_n \\ F \Leftarrow P_1, P_2, \ldots, P_m \\ \hline F \Leftarrow Q_1, Q_2, \ldots, Q_n, P_1, P_2, \ldots, P_m \end{array} \quad \text{alors}$$

Les calculs ainsi présentés font apparaître les clauses opérandes au dessus de la ligne de points, et la clause résultante en dessous d'elle.

Le processus de démonstration est rarement linéaire. Les schémas sont appliqués dans l'ordre aux clauses. Les clauses résultantes sont ajoutées à la base de connaissances. En cas d'impasse (plus de schéma applicable), un retour en arrière est fait (backtracking), pour recommencer l'application des schémas aux autres clauses.

3 Calcul des prédicats

Le calcul des prédicats est une extension du calcul des propositions. Appelé également logique du premier ordre (LPO), ou logique des prédicats, il est à la base des langages de programmation logique comme PROLOG par exemple. La puissance d'expression du calcul des prédicats repose sur l'apparition des **variables, quantificateurs et fonctions**.

3.1 Termes

Les constantes, les variables et les fonctions sont appelées **termes**.

- une constante est un sous-ensemble des symboles autorisés du domaine, comme -Jean, 23.56, V2, ...-[1],

1. Les constantes "caractères" sont le plus souvent représentées par des chaînes de caractères commençant par une lettre majuscule

– une variable est un sous-ensemble de symboles autorisés du domaine qui prend des constantes comme valeurs [2],

– une fonction est une applicatin de l'ensemble des constantes dans lui-même, comme - Nom(Jean), SquareRoot(16), ...-

3.2 Les prédicats

Un prédicat est l'application de l'ensemble des constantes dans l'ensemble (VRAI, FAUX) [3]. Par exemple,$EST_DANS(Boite, y)$, est un prédicat à deux variables. La première est instanciée par Boite, tandis que la seconde est libre. Une variable peut être instanciée par une liste de constantes. Les formules atomiques sont des combinaisons de prédicats avec leurs arguments. Les littéraux sont à la fois les formules atomiques et leur négation.

3.3 Quantificateurs

Les quantificateurs sont ceux habituellement utilisés en mathématiques :

– $\exists$ comme dans $[\exists x, OISEAU(x) \cap VERT(x)]$

– $\forall$ comme dans $[\forall x, OISEAU(x) \rightarrow PLUME(x)]$

L'ordre des quantifieurs est très important pour l'évaluation d'une expression. Par exemple :

$$[\forall x, \exists y, CONNAIT(x, y)] \tag{15}$$

qui signifie : " Toute (personne) x connait au moins une autre (personne) y", très différent de :

$$[\exists x, \forall y, CONNAIT(x, y)] \tag{16}$$

qui signifie : " Il existe une (personne) x qui connait toutes les (personnes) y".

Les variables utilisées avec un quantificateurs sont dites **liées** et les autres sont **libres**.

3.4 Formules bien formées

Les formules bien formées [4] sont définies récursivement. Les littéraux sont des formules bien formées. Le diagramme syntaxique d'une formule bien formée est le suivant :

2. Les noms de variables commencent le plus souvent par une lettre minuscule
3. En langue naturelle, un prédicat est le plus souvent exprimé par un verbe
4. Well-formed formula en anglais

3.5 Notion de clause

La démonstration de théorème utilise des expressions logiques mises sous forme clausale (sous forme de clauses). Une clause est une formule bien formée consistant en des disjonctions de littéraux. Un algorithme a été proposé par Winston pour construire des clauses à partir de formules logiques :

1. Eliminer les implications
2. Deplacer les négations dans les formules atomiques
3. Eliminer les quantificateurs existentiels
4. Renommer les variables si nécessaire
5. Deplacer les quantificateurs universels vers la gauche
6. Deplacer les disjonctions dans les littéraux
7. Eliminer les conjonctions
8. Renommer les variables, si nécessaire
9. Eliminer les quantificateurs universels

Reprenons un exemple souvent cité dans les ouvrages : *Etre un terrien implique trois choses :*

1. *le terrien vit sur la Terre*
2. *il n'existe pas de Martien qui soit parent avec un Terrien*
3. *il n'existe pas de non Terrien qui soit connu d'un Terrien*

On peut reformuler cet énoncé de la manière suivant en formules logiques :

$$\begin{array}{ll}[\forall x, & TERRIEN(x) \rightarrow [[\exists y, VIVRE\_SUR(x,y) \cap TERRE(y)] \\ & [\neg\exists y, PARENT(x,y) \cap MARTIEN(y)] \\ & [\forall y, \neg TERRIEN(y) \rightarrow \neg CONNU(x,y)]]]\end{array} \tag{17}$$

Elimination des implications

$$\begin{array}{ll}[\forall x, & \neg TERRIEN(x) \cup [[\exists y, VIVRE\_SUR(x,y) \cap TERRE(y)] \\ & [\neg\exists y, PARENT(x,y) \cap MARTIEN(y)] \\ & [\forall y, TERRIEN(y) \cup \neg CONNU(x,y)]]]\end{array} \tag{18}$$

Déplacer les négations dans les formules atomiques

$$\begin{array}{ll}[\forall x, & \neg TERRIEN(x) \cup [[\exists y, VIVRE\_SUR(x,y) \cap TERRE(y)] \\ & [\forall y, \neg PARENT(x,y) \cup \neg MARTIEN(y)] \\ & [\forall y, TERRIEN(y) \cup \neg CONNU(x,y)]]]\end{array} \tag{19}$$

Elimination des quantificateurs existentiels Dans cette phase, les fonctions dites de **skolem** doivent être introduites. Une fonction de skolem sert à substituer une variable quantifiée existentiellement. Elle s'exprime en fonction de variables définies par les quantificateurs universels de l'experession dans laquelle la substitution doit être faite. Ici, la fonction skolem de la variable existentielle y sera notée PLANETE(x), soit :

$$\begin{array}{ll}[\forall x, & \neg TERRIEN(x) \cup [[\exists y, VIVRE\_SUR(x, PLANETE(x) \cap TERRE(PLANETE(x))] \\ & [\forall y, \neg PARENT(x, y) \cup \neg MARTIEN(y)] \\ & [\forall y, TERRIEN(y) \cup \neg CONNU(x, y)]]]\end{array} \tag{20}$$

Renommer les variables, si nécessaire

$$\begin{array}{ll}[\forall x, & \neg TERRIEN(x) \\ & \cup[[VIVRE\_SUR(x, PLANETE(x) \cap TERRE(PLANETE(x))] \\ & [\forall y, \neg PARENT(x, y) \cup \neg MARTIEN(y)] \\ & [\forall z, TERRIEN(z) \cup \neg CONNU(x, z)]]]\end{array} \tag{21}$$

Déplacer les quantificateurs universels vers la gauche

$$\begin{array}{ll}[\forall x, \forall y, \forall z, & \neg TERRIEN(x) \\ & \cup[[VIVRE\_SUR(x, PLANETE(x) \cap TERRE(PLANETE(x))] \\ & [\neg PARENT(x, y) \cup \neg MARTIEN(y)] \\ & [TERRIEN(z) \cup \neg CONNU(x, z)]]]\end{array} \tag{22}$$

Déplacer les disjonctions dans les littéraux

$$\begin{array}{ll}[\forall x, \forall y, \forall z & \\ & [\neg TERRIEN(x) \cup VIVRE\_SUR(x, PLANETE(x))] \\ & \cap[\neg TERRIEN(x) \cup TERRE(PLANETE(x))] \\ & [\neg TERRIEN(x) \cup \neg PARENT(x, y) \cup \neg MARTIEN(y)] \\ & [\neg TERRIEN(x) \cup TERRIEN(z) \cup \neg CONNU(x, z)]]\end{array} \tag{23}$$

Eliminer les conjonctions

$$\begin{array}{l}[\forall x, \neg TERRIEN(x) \cup [VIVRE\_SUR(x, PLANETE(x)] \\ [\forall x, \neg TERRIEN(x) \cup TERRE(PLANETE(x))] \\ [\forall x, \forall y, \neg TERRIEN(x) \cup \neg PARENT(x, y) \cup \neg MARTIEN(y)] \\ [\forall x, \forall z, \neg TERRIEN(x) \cup TERRIEN(z) \cup \neg CONNU(x, z)]\end{array} \tag{24}$$

Renommer les variables si nécessaire

$$\begin{array}{l}[\forall x, \neg TERRIEN(x) \cup [VIVRE\_SUR(x, PLANETE(x)] \\ [\forall u, \neg TERRIEN(u) \cup TERRE(PLANETE(u))] \\ [\forall v, \forall y, \neg TERRIEN(v) \cup \neg PARENT(v, y) \cup \neg MARTIEN(y)] \\ [\forall w, \forall z, \neg TERRIEN(w) \cup TERRIEN(z) \cup \neg CONNU(w, z)]\end{array} \tag{25}$$

Eliminer les quantificateurs universels

$$\begin{array}{l} \neg TERRIEN(x) \cup [VIVRE\_SUR(x, PLANETE(x) \\ \neg TERRIEN(u) \cup TERRE(PLANETE(u)) \\ \neg TERRIEN(v) \cup \neg PARENT(v, y) \cup \neg MARTIEN(y) \\ \neg TERRIEN(w) \cup TERRIEN(z) \cup \neg CONNU(w, z) \end{array} \tag{26}$$

Ceci permet la démonstration de théorème.

3.6 Algorithme de résolution

1. Nier le théorème à prouver et ajouter le résultat dans la liste des axiomes
2. Mettre la liste des axiomes sous la forme de clauses
3. Jusqu'à ce que la liste de clauses soit vide (NIL) soit générée ou qu'il n'y ait plus de paires de clauses résolvables, chercher les clauses résolvables, les résoudre, et les ajouter à la liste des clauses
4. Si la clause vide est générée, annoncer que le théorème est vrai, s'il n'y a pas de clause résolvable, annoncer que le théorème est faux.

FIG. 7 – *Algorithme de démonstration de théorème*

4 Représentation des connaissances par objets

Imaginer comment représenter les connaissances est une gageure. La modélisation de la mémoire humaine a inspiré les psychologues, qui cherchent à reproduire les fonctions et mécanismes de raisonnement pour les étudier. Les réseaux sémantiques sont issus de ces travaux et sont devenus depuis des mécanismes récupérés par la communauté informatique pour exprimer commodément ce qui "est su" en général : "le sens commun".

4.1 Graphes et réseaux sémantiques

Les entités présentent dans un réseau sémantique sont respectivement des noeuds (représentant des objets) et des arcs (représentant des relations entre ces objets). Le lecteur s'aperçoit bien maintenant des racines des différentes méthodes d'analyse et conception "objet"!

Un graphe représentant un réseau sémantique pourrait être :

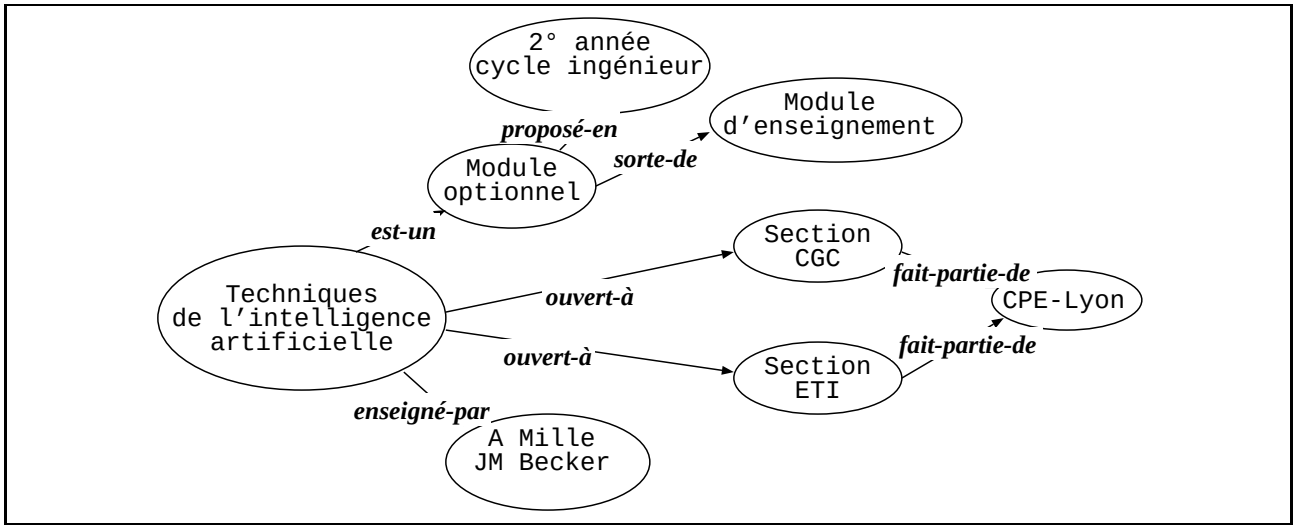

FIG. 8 – *Exemple de réseau sémantique*

Les réseaux sémantiques ont été utilisés pour l'explication (ou la compréhension) de phrases, comme :

Le professeur lit l'exercice

Le réseau sémantique correspondant :

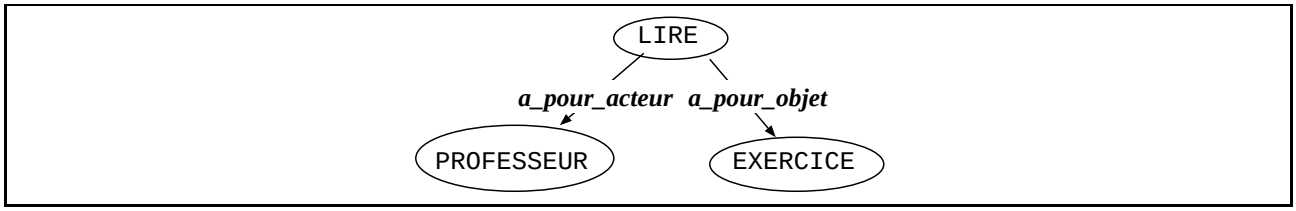

FIG. 9 – *Réseau équivalent à la phrase*

Cette phrase pourrait aussi s'exprimer sous la forme du prédicat :

[LIRE(PROFESSEUR,EXERCICE)]

Plus généralement tout objet pourra être représenté avec ses relations avec les autres selon une structure de réseau sémantique particulière appelée structure d'objet.

Un objet est défini par un ensemble de propriétés. Ces propriétés sont classiquement appelées des attributs. Une classe est un ensemble d'objets qui partagent des caractères communs.

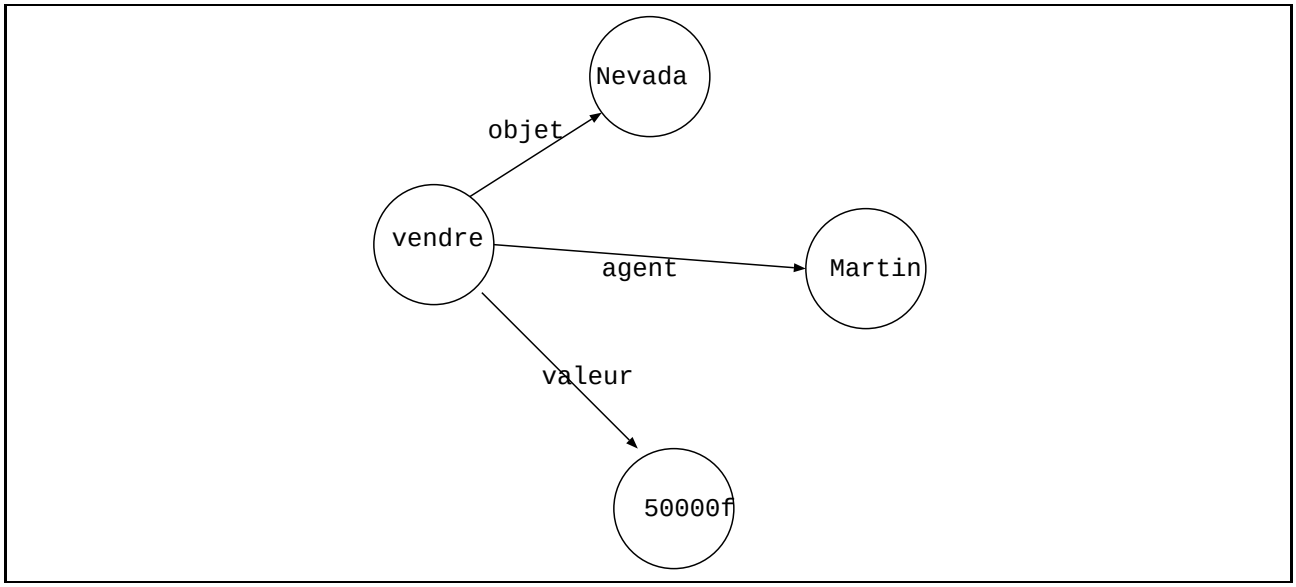

FIG. 10 – *Exemple de réseau sémantique*

L'exemple présenté ci-dessus présente le réseau sémantique correspondant à la phrase : **Martin vend une Nevada pour 50 000francs**

On peut faire une équivalence entre réseau sémantique et calcul de prédicat.

4.2 Réseau sémantique et calcul de prédicat

La représentation du réseau présenté en figure 10 peut s'exprimer sous la forme suivante :

$$Vendre(Martin, Nevada, 50000f) \tag{27}$$

Si on généralise la proposition :

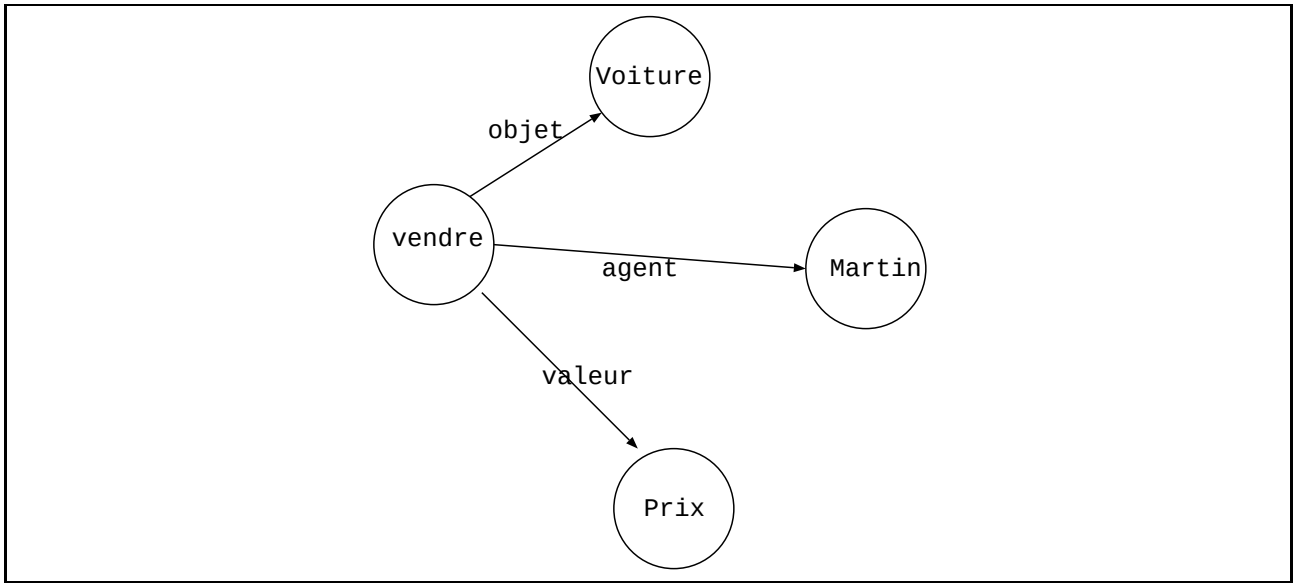

FIG. 11 – *Réseau plus générale*

Si on exprime sous la forme d'un prédicat :

$$(\exists x)(Vendre(Martin, x, y)\&objet(x)\&valeur(y)) \tag{28}$$

4.3 Propriétés d'héritage

4.3.1 Héritage simple de valeurs

L'héritage de valeur signifie qu'un objet qui hérite d'une classe (la classe peut être vue comme un objet à son niveau) voit un de ses attributs prendre la valeur que cet attribut possède dans une classe dont il hérite. Les relations d'héritage classique sont étiquetées : EST_UN lorsqu'il s'agit d'une instance (une feuille du graphe, qui n'a pas d'héritier possible) ou plus généralement SORTE_DE. La figure 12 illustre cette relation d'héritage.

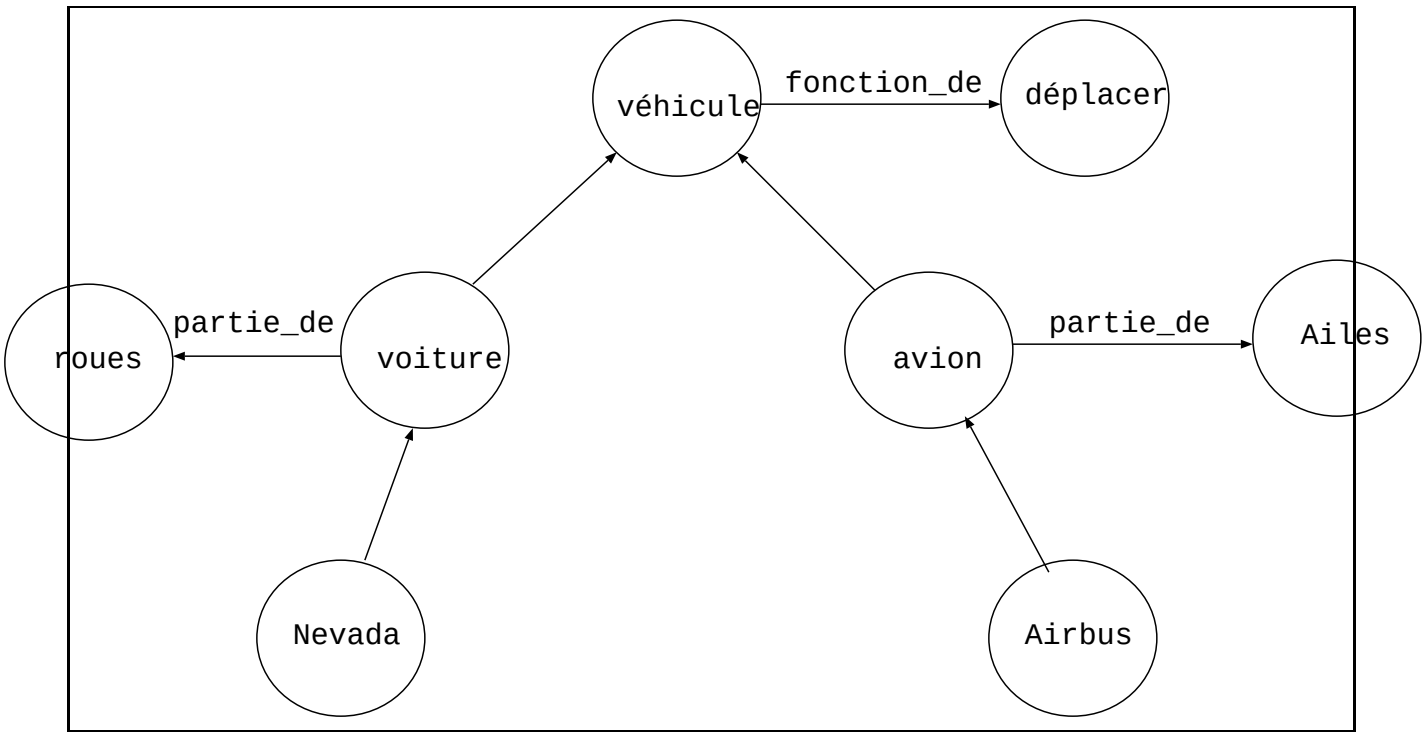

FIG. 12 – *réseau sémantique avec héritage*

L'airbus a pour valeurs héritées de posséder des ailes et d'avoir pour fonction de déplacer.

4.3.2 Héritage de calculs procéduraux

Les valeurs ne sont plus héritées directement par l'intermédiaire de l'examen de la hiérarchie d'héritage mais par le calcul d'une fonction attachée à l'attribut. Cette fonction est souvent dépendante des autres propriétés de l'objet.

4.3.3 Héritage de valeurs par défaut

Il est courant d'avoir absolument besoin d'une valeur pour procéder à des calculs ou des inférences. Une valeur par défaut peut alors être attachée à l'attribut.

Ces deux dernières façons de construire une valeur se combinent avec la première. En effet, on peut très bien hériter simplement d'une fonction de calcul ou d'une valeur par défaut. En synthèse : Un objet est caractérisé par des attributs qui possèdent plusieurs facettes possibles :

- une facette valeur qui représente la valeur effective de l'attribut. Il s'agit en fait du domaine possible de la valeur.
- une facette procédurale qui définit éventuellement calculer la valeur si le besoin s'en fait sentir

– une facette de valeur par défaut éventuelle.

D'autres facettes sont imaginables et ont été imaginées dans des langages de représentation de connaissance orientés objet.

Un objet défini de cette façon est appelé un schéma (FRAME) depuis la proposition faite par Marvin Minsky. La correspondance avec le vocabulaire américain est le suivant :

FRAME	SCHEMA	structure de base pour représenter l'objet
SLOT	ATTRIBUT	nom de propriété (relation) de l'objet
FACET	FACETTE	nom de la facette de la propriété
value	valeur	valeur prise par la propriété selon cette facette

La figure 13 illustre un tel schéma sous forme de liste.

```
(Nevada
   (Sorte_de (VALEUR Voiture))
   (Type (VALEUR Break)))

(Voiture
   (Nb_de_portes (DEFAUT 4))
   (Type (DEFAUT Berline)))
```

FIG. 13 – *Déclaration en liste du shéma nécessaire à la définion de Nevada*