Logique des propositions et logique des prédicats
Notes de cours

Jérôme Champavère

Version du 7 mars 2007

Table des matières

Introduction

La logique classique sert à exprimer des énoncés auxquels on attribue une valeur dite de **vérité** : un énoncé est soit **vrai** soit **faux** et il n'y a pas d'autre valeur possible. Par exemple quand je dis que « la Terre est plate », on peut sans problème attribuer la valeur faux à cet énoncé.

En logique classique, c'est-à-dire la logique propositionnelle et la logique du premier ordre, on ne s'intéresse pas à la dimension temporelle ou spatiale, ni à l'incertitude d'un énoncé. En d'autres temps, affirmer « la Terre est plate » était vrai ! On parle de la vérité compte tenu d'un **monde possible**. Par exemple, « il fait beau » est vrai (ou faux ?) ici et maintenant, c'est-à-dire dans le monde que je suis en train de considérer, mais dans un autre monde (un autre lieu et/ou un autre moment), cet énoncé peut avoir une valeur contraire.

1 Logique des propositions

Dans la logique propositionnelle, on étudie les relations entre des énoncés, que l'on va appeler propositions ou encore des **formules**. Ces relations peuvent être exprimées par l'intermédiaire de **connecteurs logiques** qui permettent, par composition, de construire des formules syntaxiquement correctes. On trouve principalement : la conjonction, la disjonction (inclusive), l'implication, l'équivalence et la négation.

1.1 Syntaxe

S'intéresser à la syntaxe de la logique propositionnelle, c'est considérer les formules qui sont "bien écrites". Pour cela, on se donne un alphabet, i.e. un ensemble de symboles, avec :

- un ensemble $V = \{p, q, r, \ldots\}$ dénombrable de lettres appelées **variables propositionnelles**. Il s'agit des propositions atomiques telles que par exemple « 6 est divisible par 2 ».
- les **constantes** vrai et faux.
- un ensemble (fini) de **connecteurs logiques** : $\wedge$, $\vee$, $\neg$, $\rightarrow$, $\equiv$
- les parenthèses (,)

Parmi les mots que l'on peut écrire avec cet alphabet, on va regarder ceux qui correspondent à des **expressions logiques bien formées**, autrement dit les formules que l'on a évoquées en introduction, et que l'on définit (inductivement) ainsi :

1. toutes les propositions atomiques, i.e. $p, q, r, \ldots$, sont des expressions bien formées ;
2. si A est une expression bien formée, alors $\neg A$ est une expression bien formée ;
3. si A et B sont deux expressions bien formées, alors $(A \wedge B)$, $(A \vee B)$, $(A \rightarrow B)$ et $(A \equiv B)$ sont des expressions bien formées ;
4. il n'y a pas d'autres expressions bien formées que celles des règles précédentes.

Par exemple $(((\neg p \equiv q) \vee \neg(r \wedge s)) \rightarrow q)$ est une expression bien formée, tandis que $p\neg qr \rightarrow t(\equiv$ ne l'est pas. Dans la suite, on ne considère que des expressions bien formées.

1.2 Sémantique

S'intéresser à la sémantique de la logique propositionnelle, c'est déterminer la valeur de vérité d'un énoncé, c'est-à-dire d'une formule, dans le cadre d'un de ses mondes possibles. On parle de l'**interprétation** d'une formule : il s'agit plus concrètement d'affecter une valeur vrai ou faux à chacune des variables propositionnelles qui la compose. Pour une formule à n variables, il y a 2^n mondes possibles.

1.2.1 Tables de vérité

On utilise ce qu'on appelle des **tables de vérité** pour réaliser cette interprétation. Si p et q sont des propositions, on a :

Négation

p	$\neg p$
vrai	faux
faux	vrai

Conjonction

p	q	$(p \wedge q)$
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	faux

Prenons par exemple pour p la proposition « 6 est divisible par deux » et pour q la proposition « 4 est un nombre premier ». Dans notre monde la première affirmation est vraie, la seconde est fausse. L'interprétation $I(p)$ vaudra donc vrai et $I(q)$ vaudra faux. Si l'on regarde l'interprétation de la conjonction de ces deux propositions $I(p \wedge q)$, cela revient à considérer $I(p) \wedge I(q)$, soit vrai $\wedge$ faux. La table de vérité nous indique que le résultat est faux.

Disjonction inclusive

p	q	$(p \vee q)$
vrai	vrai	vrai
vrai	faux	vrai
faux	vrai	vrai
faux	faux	faux

La disjonction inclusive correspond au « et/ou » de la langue. Il suffit que l'une des deux propositions soit vraie pour que les deux ensembles le soient.

Implication

p	q	$(p \rightarrow q)$
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	vrai
faux	faux	vrai

On reviendra sur l'implication.

Équivalence

p	q	$(p \equiv q)$
vrai	vrai	vrai
vrai	faux	faux
faux	vrai	faux
faux	faux	vrai

1.2.2 Exemple

Dans quelle monde la formule $(p \rightarrow (r \wedge s))$ est-elle vraie ?

p	r	s	$(r \wedge s)$	$(p \rightarrow ((r \wedge s))$
vrai	vrai	vrai	vrai	vrai
vrai	vrai	faux	faux	faux
vrai	faux	vrai	faux	faux
vrai	faux	faux	faux	faux
faux	vrai	vrai	vrai	vrai
faux	vrai	faux	faux	vrai
faux	faux	vrai	faux	vrai
faux	faux	faux	faux	vrai

$(p \rightarrow (r \wedge s))$ est vrai quand p, r et s sont tous les trois vrais ou quand p est faux quelles que soient les valeurs de r et s.

1.2.3 Remarque sur l'implication

Dans le langage courant, l'implication est souvent traitée comme « si une hypothèse H est vérifiée, alors on observe la conclusion C » ou bien on dit plus simplement que « H implique C ». En mathématiques on dirait que H est une condition suffisante de C. Une phrase telle que « si 2+2=5, alors Tokyo est la capitale du Japon » n'a pas beaucoup de sens pour nous, pourtant cet énoncé est vrai du point de vue de la logique propositionnelle ! En effet, on ne s'intéresse pas à la véracité de l'hypothèse.

En résumé, en logique propositionnelle, l'énoncé « H implique C » est vrai dans le cas où C est vrai et dans le cas où H est faux. Pour s'en convaincre, il faut simplement constater que l'unique cas raisonnable pour que « H implique C » soit faux est lorsque H est vrai et C est faux. Dans tous les autres cas, la règle est que l'énoncé est vrai.

1.3 Consistance, inconsistance et validité

1.3.1 Définitions

Modèle On appelle **modèle** une interprétation pour laquelle une formule est vraie. Par exemple $\langle p = \mathsf{faux}, r = \mathsf{vrai}, s = \mathsf{faux}\rangle$ est un modèle de $(p \to (r \wedge s))$.

Consistance On dit qu'une formule A est **consistante**, ou **satisfiable**, s'il existe une interprétation de ses variables propositionnelles qui la rende vraie. Autrement dit A est consistante si et seulement si A a un modèle. $(p \to (r \wedge s))$ est une formule consistante.

Inconsistance Une formule pour laquelle il n'existe pas d'interprétation qui la rende vraie est dite **inconsistante**, ou encore **insatisfiable**, ou plus simplement **fausse**. $(p \wedge \neg p)$ est inconsistante :

p	$\neg p$	$(p \wedge \neg p)$
vrai	faux	faux
faux	vrai	faux

Validité Si une formule A est vraie pour n'importe quelle interprétation de ses variables propositionnelles, on dit qu'elle est **valide**. On dit encore que A est une **tautologie** et on le note $\models A$. $(p \vee \neg p)$ est une tautologie :

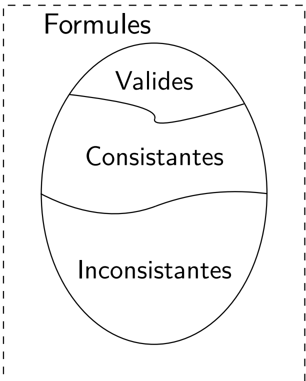
p	$\neg p$	$(p \vee \neg p)$
vrai	faux	vrai
faux	vrai	vrai

Il s'agit ici du principe du "tiers-exclu" dont on se sert implicitement en mathématiques pour faire un raisonnement par l'absurde. Il existe d'autres tautologies dont on se sert fréquemment :

$\models (A \to A)$ et $\models (A \equiv A)$	identité
$\models (A \vee \neg A)$	tiers-exclu
$\models \neg(A \wedge \neg A)$	non contradiction
$\models (A \equiv \neg\neg A)$	double négation

En logique classique, toutes les tautologies sont équivalentes. Ce n'est pas forcément le cas dans des logiques non classiques où ces principes ne sont pas nécessairement admis.

En résumé :



1.3.2 Équivalences logiques

Il existe un certain nombre de tautologies utilisées en calcul propositionnel qui sont parfois qualifiées **d'équivalences logiques** standards. Elles sont présentées ci-dessous.

$$
\begin{array}{ll}
\models [(A \wedge B) \equiv (B \wedge A)] & \text{commutativité} \\
\models [(A \vee B) \equiv (B \vee A)] & \ldots \\
\models [(A \equiv B) \equiv (B \equiv A)] & \ldots \\
\models [((A \wedge B) \wedge C) \equiv ((A \wedge (B \wedge C))] & \text{associativité} \\
\models [((A \vee B) \vee C) \equiv (A \vee (B \vee C))] & \ldots \\
\models [((A \equiv B) \equiv C) \equiv (A \equiv (B \equiv C))] & \ldots
\end{array}
$$

$$
\begin{array}{ll}
\models [((A \wedge B) \vee C) \equiv ((A \vee C) \wedge (B \vee C))] & \text{distributivité} \\
\models [((A \vee B) \wedge C) \equiv ((A \wedge C) \vee (B \wedge C))] & \ldots \\
\models [(A \wedge (A \vee B)) \equiv A] & \text{absorption} \\
\models [(A \vee (A \wedge B)) \equiv A] & \ldots \\
\models [(A \wedge A) \equiv A] & \text{idempotence} \\
\models [(A \vee A) \equiv A] & \ldots
\end{array}
$$

$$
\begin{array}{ll}
\models [\neg(A \wedge B) \equiv (\neg A \vee \neg B)] & \text{lois de Morgan} \\
\models [\neg(A \vee B) \equiv (\neg A \wedge \neg B)] & \ldots \\
\models [(A \rightarrow B) \equiv (\neg A \vee B)] & \text{implication*} \\
\models [(A \rightarrow B) \equiv \neg(A \wedge \neg B)] & \ldots \\
\models [(A \equiv B) \equiv ((A \rightarrow B) \wedge (B \rightarrow A))] & \text{équivalence} \\
\models [(A \equiv B) \equiv ((A \wedge B) \vee (\neg A \wedge \neg B))] & \ldots
\end{array}
$$

* Table de vérité prouvant cette équivalence logique :

A	B	$(A \rightarrow B)$	$\neg A$	$(\neg A \vee B)$
vrai	vrai	vrai	faux	vrai
vrai	faux	faux	faux	faux
faux	vrai	vrai	vrai	vrai
faux	faux	vrai	vrai	vrai

$\models [(A \to B) \equiv (\neg B \to \neg A)]$	contraposition
$\models [(A \to (B \to C)) \equiv ((A \to B) \to (A \to C))]$	autodistributivité
$\models [(A \to (B \to C)) \equiv (B \to (A \to C))]$	import-export
$\models [((A \to B) \land (B \to C)) \equiv (A \to C)]$	transitivité

1.4 Conséquence logique

Jusqu'à présent, nous nous sommes concentrés sur l'aspect de représentation des connaissances en terme de logique propositionnelle. En logique, et plus spécifiquement en Intelligence Artificielle, on est intéressé par des procédures de raisonnement. La question centrale est : à partir d'un certain nombre de connaissances, que peut-on déduire ? Et comment peut-on le déduire ? Des questions importantes s'ensuivent : le principe selon lequel je raisonne est-il correct — ce que j'ai déduit est "vraiment vrai" — et est-il complet — suis-je sûr de pouvoir tout déduire ?

Le principe du raisonnement logique est la relation de conséquence logique entre les énoncés, l'idée qu'un énoncé découle logiquement d'un autre énoncé. Par exemple vous savez qu'à la dernière séance du cours vous aurez un contrôle. Vous savez également qu'en général un contrôle aboutit à une note. Vous en déduisez logiquement que vous aurez une note à la fin du cours.

$$\frac{\text{Contrôle}, \text{Contrôle} \to \text{Note}}{\text{Note}}$$

1.4.1 Définition

Soit $\{F_1, \ldots, F_n\}$ un ensemble de formules. On dit qu'une formule C est une **conséquence logique** de l'ensemble $\{F_1, \ldots, F_n\}$, que l'on écrit $\{F_1, \ldots, F_n\} \models C$, si toute interprétation rendant vraies les formules F_1, ..., F_n simultanément, rend vraie la formule C. Autrement dit, tout modèle de $\{F_1, \ldots, F_n\}$ est un modèle de C. Dans le petit exemple précédent, nous avions $\{\text{Contrôle}, \text{Contrôle} \to \text{Note}\} \models \text{Note}$.

Contrôle	Contrôle $\to$ Note	Note
<u>vrai</u>	<u>vrai</u>	<u>vrai</u>
vrai	faux	faux
faux	vrai	vrai
faux	vrai	faux

1.4.2 Propriétés

La notion de conséquence logique est intimement liée à celle de tautologie. En fait si par exemple une formule A est une tautologie c'est parce qu'on peut la déduire de l'ensemble vide $\emptyset \models A$: une tautologie existe indépendamment d'un ensemble de prémisses. Plus généralement, si $\mathcal{F}$[1] est un ensemble de formules et C une formule, on a le **théorème de déduction** suivant :

$$\boxed{\mathcal{F} \models C \text{ si et seulement si } \models (\mathcal{F} \to C)}$$

Ce théorème peut s'exprimer ainsi : « C se déduit de $\mathcal{F}$ si et seulement si $(\mathcal{F} \to C)$ est une tautologie ». Parmi les corollaires intéressants de ce théorème, on retiendra essentiellement le suivant :

$$\boxed{\mathcal{F} \models C \text{ si et seulement si } \mathcal{F} \cup \{\neg C\} \text{ est inconsistant}}$$

Ce résultat, aussi connu sous le nom de **théorème de réfutation**, est à la base de la méthode de résolution que nous verrons plus loin.

1.4.3 Exemple

Considérons les arguments suivants[2].

> *Si Didier est l'auteur de ce bruit, il est stupide ou dépourvu de principes. Didier n'est ni stupide ni dépourvu de principes. Donc Didier n'est pas l'auteur de ce bruit.*

Pour nous convaincre de la validité de ce raisonnement, on le décompose comme suit en posant :

- D : « Didier est l'auteur de ce bruit »
- S : « Didier est stupide »
- P : « Didier est dépourvu de principes »
- H_1 : « Si Didier est l'auteur de ce bruit, il est stupide ou dépourvu de principes » $(D \to (S \vee P))$
- H_2 : « Didier n'est pas stupide » $\neg S$
- H_3 : « Didier n'est pas dépourvu de principes » $\neg P$
- C : « Didier n'est pas l'auteur de ce bruit » $\neg D$

On pose la question : $\{H_1, H_2, H_3\} \models C$ (Peut-on déduire logiquement D de H_1, H_2 et H_3 ? La table de vérité nous permet de vérifier aisément l'assertion.

D	S	P	$H_1 : (D \to (L \vee P))$	$H_2 : \neg S$	$H_3 : \neg P$	$C : \neg D$
faux	faux	faux	<u>vrai</u>	<u>vrai</u>	<u>vrai</u>	<u>vrai</u>
faux	faux	vrai	vrai	vrai	faux	vrai
faux	vrai	faux	vrai	faux	vrai	vrai
faux	vrai	vrai	vrai	vrai	faux	vrai
vrai	faux	faux	faux	vrai	vrai	faux
vrai	faux	vrai	vrai	faux	faux	faux
vrai	vrai	faux	vrai	vrai	vrai	faux
vrai	vrai	vrai	vrai	vrai	faux	faux

[1] Nous préciserons ultérieurement à quoi correspond ce $\mathcal{F}$.

[2] Pour la Science, Sept. 1986.

La table de vérité est donc un moyen sûr pour contrôler la validité d'une déduction logique. Néanmoins, comme nous l'avons déjà signalé, son principale inconvénient réside dans le fait que c'est une méthode inefficace en pratique car pour trouver un modèle nous pouvons énumérer jusqu'à 2^n interprétations, n étant le nombres de variables propositionnelles présents dans l'ensemble de formules considéré.

1.5 Principe de résolution

La conséquence logique telle que nous l'avons vue jusqu'ici à été présentée de manière sémantique. En effet, nous avons utilisé les tables de vérité pour interpréter un ensemble de formules logiques et vérifier qu'on pouvait en déduire une autre dont la valeur est vraie dans les modèles où chacun des prémisses est consistant.

La déduction logique peut en réalité très bien s'effectuer sans connaître les valeurs de vérité associées aux variables propositionnelles que l'on manipule. Pour cela, on utilise des méthodes purement syntaxiques, c'est-à-dire manipulant directement les symboles de la logique au moyen d'un système à base de règles. Les objets dérivés par l'application de ces règles sont appelés **théorèmes**, noté $\vdash A$ (A est un théorème). On montre que dans un tel système correct et complet, les théorèmes sont exactement les tautologies :

Si $\vdash A$ alors $\models A$	correction
Si $\models A$ alors $\vdash A$	complétude

Pour résoudre un problème de logique, on peut appliquer différents **schémas de raisonnement**, ou **règles d'inférence**. Dans l'exemple {Contrôle, Contrôle $\rightarrow$ Note} $\models$ Note, le schéma utilisé s'appelle *modus ponens*.

Nous aurions pu raisonner différement, en partant de la négation de la conclusion, c'est-à-dire ¬Note, associée à notre connaissance Contrôle $\rightarrow$ Note. En utilisant la règle dite du *modus tollens*, nous avons {Contrôle $\rightarrow$ Note, ¬Note $\models$ ¬Contrôle}. Cela étant contradictoire avec le fait que nous savons Contrôle, nous en déduisons nécessairement Note.

$$\frac{\text{Contrôle} \rightarrow \text{Note}, \neg\text{Note}}{\neg\text{Contrôle}}$$

Parmi les divers schémas de raisonnement existants, nous allons nous concentrer sur la **règle de résolution**, dont le *modus ponens* et le *modus tollens* sont en fait des cas particuliers. Cette règle d'inférence s'exprime ainsi :

$$\boxed{\frac{X \vee A, \neg X \vee B}{A \vee B}}$$

Pour illustrer ce schéma, considérons l'ensemble suivant de propositions : {« Julien est à la Maison ou il est au Cinéma »,« Julien n'est pas à la Maison ou il est au Travail »}. Si Julien est à la maison, « il est au Travail » est nécessairement vrai ; de même s'il n'est pas à la maison, « il est au Cinéma » est nécessairement vrai. Donc il est nécessairement au cinéma ou au travail.

1.5.1 Forme clausale

Le principe de résolution s'applique plus généralement à un ensemble de clauses.

Définitions

- Un **littéral** est une variable propositionnelle (par ex. p) ou la négation d'une variable propositionnelle (par ex. $\neg p$).
- Une **clause** est une disjonction de littéraux (par ex. $p \vee \neg q \vee r$). On note $\perp$ la clause vide.
- Un **ensemble de clauses** est la conjonction de toutes les clauses qu'il contient (par ex. $\{\neg p \vee q \vee \neg r, p \vee q, \neg p \vee r\}$ est le conjonction de disjonctions $(\neg p \vee q \vee \neg r) \wedge (p \vee q) \wedge (\neg p \vee r)$).
- La **résolvante** de deux clauses correspond à la clause résultant de l'application de la règle de résolution sur ces deux clauses. Dans notre exemple précédent, Cinéma $\vee$ Travail est la résolvante de Maison $\vee$ Cinéma et $\neg$Maison $\vee$ Travail. Dans le cas général, on aura :

$$\frac{x_1 \vee \cdots \vee x_i \vee \cdots \vee x_n, \qquad y_1 \vee \cdots \vee y_j \vee \cdots \vee y_m}{x_1 \vee \cdots \vee x_{i-1} \vee x_{i+1} \vee \cdots \vee x_n \vee y_1 \vee \cdots \vee y_{j-1} \vee y_{j+1} \vee \cdots \vee y_m}$$

  où x_i et y_j sont des littéraux complémentaires (par exemple p et $\neg p$).

La calcul de la résolvante est l'unique règle d'inférence pour le principe de résolution. Nous constatons que cette règle ne s'applique que sur une forme bien particulière de formules, la clause, qui correspond à une disjonction de littéraux. Nous avons observés en TD que toutes les formules logiques peuvent s'écrire en utilisant uniquement un nombre restreint de connecteurs.

Propriété Pour résoudre un problème de logique avec le principe de résolution, nous allons donc devoir transformer les énoncés et les mettre sous la forme de conjonctions de clauses, qui ne sont composées que de la négation ($\neg$), de la disjonction ($\vee$) et de la conjonction ($\wedge$). C'est ce que l'on appelle la mise sous **forme normale conjonctive** d'une formule logique, rendue possible par la propriété ci-dessous.

$\boxed{\text{Toute formule admet une forme clausale qui lui est équivalente.}}$

Par exemple, si l'on reprend l'énoncé « Si Didier est l'auteur de ce bruit, il est stupide ou dépourvu de principes », exprimé par la formule ($D \rightarrow$

$(S \vee P))$, sa forme normale conjonctive sera l'unique clause $\neg D \vee S \vee P$. On utilise simplement le fait que $(A \rightarrow B) \equiv (\neg A \vee B)$.

1.5.2 Méthode de transformation

Pour mettre une formule sous forme clausale, on s'appuie sur les équivalences logiques standards que nous avons déjà vues, en suivant les étapes de l'algorithme suivant.

1. Élimination de $\equiv$: remplacement de $(X \equiv Y)$ par $(X \rightarrow Y) \wedge (Y \rightarrow X)$.
2. Élimination de $\rightarrow$: remplacement de $(X \rightarrow Y)$ par $(\neg X \vee Y)$.
3. La forme normale conjonctive exige que la négation $\neg$ n'apparaissent que dans les littéraux. On utilise donc l'annulation de la double négation ainsi que les régles de réécriture dérivées des lois de Morgan :
   – remplacement de $\neg\neg X$ par X ;
   – remplacement de $\neg(X \wedge Y)$ par $(\neg X \vee \neg Y)$;
   – remplacement de $\neg(X \vee Y)$ par $(\neg X \wedge \neg Y)$.
4. Des trois étapes précédentes, on obtient une expression ne contenant plus que des $\wedge$ et des $\vee$ imbriqués. On applique auntant de fois que nécessaire les lois de distribution $(X \vee (Y \wedge Z)) \equiv ((X \vee Y) \wedge (X \vee Z))$ et $(X \wedge (Y \vee Z)) \equiv ((X \wedge Y) \vee (X \wedge Z))$.

L'expression obtenue à la fin de cette procédure est une forme clausale équivalente à la formule de départ.

Remarques

– Les clauses comportant deux littéraux opposés sont valides (tiers-exclu) et peuvent donc être supprimées (par ex. $p \vee q \vee \neg r \vee \neg q$).
– On peut aussi supprimer les répétitions d'un littéral au sein d'une même clause (par ex. $\neg p \vee q \vee \neg r \vee \neg p$ équivaut à $\neg p \vee q \vee \neg r$).
– Si dans une formule clausale une clause C_i est incluse dans une clause C_j alors la clause C_j peut être supprimée (la valeur de la conjonction des deux clauses ne dépend que de la valeur de C_i). Par exemple $C_i = \underline{p} \vee \underline{q} \vee \underline{r}$ et $C_j = \underline{p} \vee \neg s \vee t \vee \underline{q} \vee \underline{r}$.

Exemple Résolution d'une énigme par la logique des propositions.

> *Vous êtes perdus sur une piste dans le désert. Vous arrivez à une bifurcation. Chacune des deux pistes est gardée par un sphynx que vous pouvez interroger. Les pistes peuvent soit conduire à une oasis, soit se perdre dans le désert profond (au mieux, elle conduisent toutes à une oasis, au pire elles se perdent toutes les deux).*
>
> *A. Le sphynx de droite vous répond : « Une au moins des deux pistes conduit à une oasis. »*

B. Le sphynx de gauche vous répond : « La piste de droite se perd dans le désert. »

C. Vous savez que les sphynx disent tous les deux la vérité, ou bien mentent tous les deux.

On pose :

D : « Il y a une oasis au bout de la route de droite. »

G : « Il y a une oasis au bout de la route de gauche. »

On a :

1. $A = D \vee G$
2. $B = \neg D$
3. $A \equiv B$ (formule $\mathcal{F}$ qu'on cherche à vérifier)

Mise sous forme normale conjonctive :

$$\begin{aligned}
\mathcal{F} &= A \equiv B \\
&= (A \rightarrow B) \wedge (B \rightarrow A) \\
&= ((D \vee G) \rightarrow \neg D) \wedge (\neg D \rightarrow (D \vee G)) \\
&= (\neg(D \vee G) \vee \neg D) \wedge (\neg\neg D \vee (D \vee G)) \\
&= ((\neg D \wedge \neg G) \vee \neg D) \wedge (D \vee (D \vee G)) \\
&= (\neg D \vee \neg D) \wedge (\neg G \vee \neg D) \wedge (D \vee D \vee G) \\
&= (\neg D) \wedge (\neg D \vee \neg G) \wedge (D \vee G) \\
&= (\neg D) \wedge (D \vee G)
\end{aligned}$$

On aboutit à l'ensemble de clauses $\mathcal{C} = \{\neg D, D \vee G\}$. L'application de la règle de résolution nous indique que la route de gauche conduit effectivement à une oasis :

$$\frac{\neg D, D \vee G}{G}$$

1.5.3 Résolution par réfutation

Pour résoudre un problème de logique par la méthode de résolution, on s'appuie sur le théorème de réfutation que nous avons présenté plus haut, à savoir :

Une formule C est la conséquence logique d'un ensemble de formules $\mathcal{F} = \{H_1, \ldots, H_n\}$ si et seulement si $\mathcal{F} \cup \{\neg C\}$ est inconsistant.

La stratégie de résolution est donc la suivante : si ce n'est pas dajà fait, on met sous forme normale conjonctive l'ensemble des prémisses, ainsi que la négation de la conséquence logique que l'on cherche à prouver. Puis, à l'aide de l'unique règle d'inférence de la résolution, on cherche à mettre en évidence une contradiction dans la formule $H_1 \wedge \cdots \wedge H_n \wedge fnc(\neg C)$. Chaque pas de calcul va enrichir l'ensemble des clauses qui constitue en fait notre base de connaissances. On peut résumer l'approche dans l'algorithme suivant.

1. Choisir deux clauses dont on peut calculer la résolvante et la calculer effectivement.
2. Si aucune des deux conditions ci-dessous n'est remplie, répéter l'étape 1.
   - Aucune nouvelle clause ne peut-être ajoutée à la base de connaissances : dans ce cas, C n'est pas une conséquence logique de $\mathcal{F}$.
   - La résolvante produite est la clause vide : dans ce cas, $\mathcal{F}$ a pour conséquence logique C.

La clause vide résulte de l'application de la règle de résolution au cas particulier où les deux clauses constituent des littéraux contraires :

$$\frac{X, \neg X}{\bot}$$

Exemple Considérons l'exemple de Didier. On dispose des connaissances suivantes : $D \to (S \lor P)$, $\neg S$ et $\neg P$. Ramené sous forme clausale, on a $\{\neg D \lor S \lor P, \neg S, \neg P\}$. On cherche à déduire $\neg D$. On en prend donc la négation, soit D, et on vérifie la consistance de $\{\neg D \lor S \lor P, \neg S, \neg P, D\}$.

$$\begin{tikzcd}[column sep=small]
\neg D \lor S \lor P \arrow[dr, no head] & & \neg S \arrow[dl, no head] & & \\
& \neg D \lor P \arrow[dr, no head] & & \neg P \arrow[dl, no head] & \\
& & \neg D \arrow[dr, no head] & & D \arrow[dl, no head] \\
& & & \bot &
\end{tikzcd}$$

Exercice Montrons $\{p \lor q \lor r, \neg p \lor q \lor r, \neg q \lor r\} \models r$ à l'aide de la méthode de résolution par réfutation.

Ensemble de départ	$\{p \lor q \lor r, \neg p \lor q \lor r, \neg q \lor r, \neg r\}$
Premier pas	$\{p \lor q \lor r, \neg p \lor q \lor r, \neg \mathbf{q} \lor \mathbf{r}, \neg \mathbf{r}, \underline{\neg q}\}$
Deuxième pas	$\{\mathbf{p} \lor \mathbf{q} \lor \mathbf{r}, \neg \mathbf{p} \lor \mathbf{q} \lor \mathbf{r}, \neg q \lor r, \neg r, \neg q, \underline{q \lor r}\}$
Troisième pas	$\{p \lor q \lor r, \neg p \lor q \lor r, \neg q \lor r, \neg r, \neg \mathbf{q}, \mathbf{q} \lor \mathbf{r}, \underline{r}\}$
Quatrième pas	$\{p \lor q \lor r, \neg p \lor q \lor r, \neg q \lor r, \neg \mathbf{r}, \neg q, q \lor r, \mathbf{r}, \underline{\bot}\}$

1.5.4 Correction et complétude

Validité de la résolution

Exemple Soit les énoncés « Julien est à la Maison ou il est au Cinéma » et « Julien n'est pas à la Maison ou il est au Travail », que l'on écrit sous forme de propositions $M \vee C$ et $\neg M \vee T$. On veut montrer $\{M \vee C, \neg M \vee T\} \models C \vee T$. D'après le théorème de déduction, cela revient à vérifier que $((M \vee C) \wedge (\neg M \vee T) \rightarrow (C \vee T))$ est valide. Nous le prouvons à l'aide d'une table de vérité.

M	C	$\neg M$	T	$(M \vee C)$	$(\neg M \vee T)$
faux	faux	faux	faux	faux	vrai
faux	faux	faux	vrai	faux	vrai
faux	vrai	faux	faux	vrai	vrai
faux	vrai	faux	vrai	vrai	vrai
vrai	faux	vrai	faux	vrai	faux
vrai	faux	vrai	vrai	vrai	vrai
vrai	vrai	vrai	faux	vrai	faux
vrai	vrai	vrai	vrai	vrai	vrai

$((M \vee C) \wedge (\neg M \vee T))$	$(C \vee T)$	$(((M \vee C) \wedge (\neg M \vee T)) \rightarrow (C \vee T))$
faux	faux	vrai
faux	vrai	vrai
vrai	vrai	vrai
vrai	vrai	vrai
faux	faux	vrai
vrai	vrai	vrai
faux	vrai	vrai
vrai	vrai	vrai

Cas général On pose $x_i = \neg y_j$:

$$\frac{x_1 \vee \cdots \vee x_i \vee \cdots \vee x_n, \qquad y_1 \vee \cdots \vee y_j \vee \cdots \vee y_m}{x_1 \vee \cdots \vee x_{i-1} \vee x_{i+1} \vee \cdots \vee x_n \vee y_1 \vee \cdots \vee y_{j-1} \vee y_{j+1} \vee \cdots \vee y_m}$$

Si x_i est vrai alors $x_1 \vee \cdots \vee x_n$ est vrai et y_j est faux, et $y_1 \vee \cdots \vee y_{j-1} \vee y_{j+1} \vee \cdots \vee y_m$ doit être vrai, sinon l'ensemble n'est pas consistant. Si x_i est faux alors y_j est vrai, donc $y_1 \vee \cdots \vee y_m$ est vrai, et $x_1 \vee \cdots \vee x_{i-1} \vee x_{i+1} \vee \cdots \vee x_n$ doit être vrai. Donc $x_1 \vee \cdots \vee x_{i-1} \vee x_{i+1} \vee \cdots \vee x_n \vee y_1 \vee \cdots \vee y_{j-1} \vee y_{j+1} \vee \cdots \vee y_m$ doit être vrai.

Complétude de la résolution On appelle **clôture par résolution** d'un ensemble de clauses E, noté $\mathcal{C}(E)$, l'ensemble de toutes les clauses qu'on peut dériver en appliquant itérativement la règle de résolution aux clauses contenues dans E ainsi qu'à leurs dérivées.

La clotûre $\mathcal{C}(E)$ est un ensemble fini : on ne peut construire qu'un nombre fini de clauses distinctes avec un nombre fini de littéraux. La taille d'une clause est limitée par le nombre de variables propositionnelles considérées dans l'ensemble E car les copies multiples de littéraux sont supprimées. Une conséquence immédiate de cette remarque est que l'algorithme termine.

Remarque La règle de résolution est complète dans un sens spécialisée. Cela signifie que l'on ne peut pas générer de nouvelles connaissances. Par exemple si l'on connaît simplement A, on ne peut pas déduire la conséquence $A \vee B$, ce qui serait pourtant correct. En effet, on a $\{A\} \models A \vee B$!

A	B	$(A \vee B)$	$(A \rightarrow (A \vee B))$
faux	faux	faux	vrai
faux	vrai	vrai	vrai
vrai	faux	vrai	vrai
vrai	vrai	vrai	vrai

En revanche, grâce au théorème de réfutation, on peut répondre à toute question, en particulier : $\{A\} \models A \vee B$? On doit montrer que l'ensemble $\{A, \neg(A \vee B)\}$ est soit inconsitant (la clause vide est produite), soit clos par résolution (toutes les clauses sont produites et on ne peut plus en produire d'autre). Dans notre exemple, l'ensemble évolue ainsi :
- $\{A, \neg A \wedge \neg B\}$ (calcul préliminaire)
- $\{A, \neg A, \neg B\}$ (forme clausale)
- $\{\mathbf{A}, \neg\mathbf{A}, \neg B, \underline{\perp}\}$ (résolution)

L'ensemble est inconsistant donc on a montré $\{A\} \models A \vee B$.

1.6 Chaînage avant et chaînage arrière

1.6.1 Clauses de Horn

Une **clause de Horn** est une disjonction de littéraux, c'est-à-dire en fait une clause dont un seul au maximum de ses littéraux est positif. Par exemple, $\neg p \vee \neg q \vee r$, $\neg s \vee \neg r$ ou encore simplement le littéral t sont des clauses de Horn, tandis que $p \vee q \vee \neg r$ n'en est pas une.

L'idée d'avoir ce genre de clauses est qu'on peut les écrire comme une conjonction de littéraux positifs impliquant un littéral positif unique. Par exemple $\neg p \vee \neg q \vee r$ est logiquement équivalent à $(p \wedge q) \rightarrow r$. La partie se situant à gauche de l'implication s'appelle la **prémisse**, et la partie se situant à sa droite la **conclusion**. Une autre façon de nommer prémisse et conclusion est la suivante : on appelle **corps** l'ensemble des littéraux négatifs de la clause, et **tête** son unique littéral positif lorsqu'il existe.

$$\overbrace{(p \wedge q)}^{\text{Corps}} \rightarrow \overbrace{r}^{\text{Tête}}$$

On appelle **clauses définies** celles ayant exactement un littéral positif. Par exemple, $\neg p \vee \neg q \vee r$ ou bien t sont définies, alors que $\neg s \vee \neg r$ ne l'est pas.

Un **fait** est une clause de Horn définie dépourvue de littéraux négatifs. Par exemple r est un fait, et $\neg p \vee \neg q \vee r$ et $\neg s \vee \neg r$ n'en sont pas.

L'intérêt de mettre une formule sous la forme d'une clause de Horn est essentiellement pratique. En effet, un grand nombre d'énoncés peuvent s'écrire uniquement en employant ce type de clauses, comme par exemple l'énoncé ci-dessous.

- *Les gens qui ont la rougeole doivent prendre le médicament X* $(r \to x)$.
- *Les gens qui ont de la fièvre et des points rouges au fond de la gorge ont la rougeole* $((f \wedge g) \to r)$.
- *Ceux pour qui la température est au-dessus de 38 ° sont considérés comme ayant de la fièvre* $(s \to f)$.
- *Armand a des points rouges au fond de la gorge et a une température de 39 ° 5* $(g \wedge s)$.

Il existe essentiellement deux algorithmes d'inférence pour réaliser des déductions sur des clauses de Horn. Ils s'appellent **chaînage avant** et **chaînage arrière**, et sont tout à fait naturels pour les humains. Ils présentent en outre l'avantage d'effectuer un calcul de résolution en temps linéaire par rapport à la taille de l'ensemble des connaissances.

Les clauses de Horn constituent donc une restriction[3] efficace dans de nombreux cas pratiques de problèmes de raisonnement en logique.

1.6.2 Chaînage avant

Le chaînage avant consiste à combiner des faits connus pour en déduire de nouveaux. Par exemple, de l'énoncé posé précédemment, peut-on déduire le fait x ? Nous aurons la chaîne d'inférences suivante :

$$\cfrac{\cfrac{\cfrac{s, \qquad s \to f}{f, g, \quad (f \wedge g) \to r}\ \text{de } s \text{ on déduit } f}{r, r \to x}\ \text{de } f \text{ et } g \text{ on déduit } r}{x}\ \text{de } r \text{ on déduit } x$$

Le principe du chaînage avant est simple : si toutes les prémisses d'une implication sont connues, on ajoute la conclusion à la base de connaissances. On peut répéter ce processus jusqu'à tant que :

- soit la requête recherchée est produite (par exemple x) ;
- soit on ne peut rien déduire de plus.

[3] Il n'est pas possible de mettre toutes les propositions sous la forme de clauses de Horn.

L'algorithme est donc identique à l'algorithme de résolution, en utilisant simplement la règle du *modus ponens*, qui n'est en fait rien d'autre qu'un cas particulier de la règle de résolution, comme nous l'avions déjà mentionné.

Dans le cadre du chaînage avant, le raisonnement est **guidé par les connaissances** déjà acquises. On peut déduire des connaissances jusqu'à un **point fixe**, c'est-à-dire un état à partir duquel on ne peut plus rien inféré de nouveau en appliquant le *modus ponens*.

1.6.3 Chaînage arrière

Dans le chaînage arrière, on part du but que l'on veut atteindre, autrement dit un fait que l'on cherche à déduire de nos connaissances. Exemple de raisonnement :

x est-il un fait connu ? Non. On considère alors les règles dans lesquelles x est une conséquence, c'est-à-dire celles où x est la tête. Nous avons $r \to x$. Il faut désormais prouver r par chaînage arrière.

- r ? $f \wedge g \to r$
- f ? $s \to f$
- s ? oui, donc f
- g ? oui, donc r

Nous avons satisfait toutes les dépendances en amont pour x par chaînage arrière, de ce fait on déduit x.

Le raisonnement dans le chaînage arrière est **guidé par le but**. Le moteur d'inférence de PROLOG fonctionne essentiellement sur ce principe.

1.7 Axiomatisation

La logique propositionnelle est une théorie. Cela signifie qu'à partir de quelques axiomes et règles de déduction, on peut déduire toutes les formules valides, autrement dit les **théorèmes**. On considère les expressions bien formées utilisant une ensemble fini de variables propositionnelles, les opérateurs $\neg$ et $\to$, ainsi que les parenthèses, et on pose les trois axiomes suivants :

1. $(p \to (q \to p))$ (conséquence de l'hypothèse)
2. $((p \to (q \to r)) \to ((p \to q) \to (p \to r)))$ (autodistributivité)
3. $((\neg p \to \neg q) \to (q \to p))$ (contraposition partielle)

On se donne comme règles de déduction la règle de substitution et le *modus ponens*.

On peut obtenir le théorème $(p \to p)$ en appliquant la démonstration ci-dessous.

(1) $(p \to (q \to p))$ *axiome 1*
(2) $(p \to ((q \to p) \to p))$ *substitution de q par $q \to p$ (axiome 1)*
(3) $((p \to ((q \to p) \to p)) \to ((p \to (q \to p)) \to (p \to p)))$ *substitution de q par $(q \to p)$ et de r par p (axiome2)*

(4)	$((p \to (q \to p)) \to (p \to p))$	modus ponens *sur les lignes (2) et (3)*
(5)	$(p \to p)$	modus ponens *sur les lignes (1) et (4)*

2 Logique des prédicats

La logique propositionnelle ne permet de décrire que des constructions simples du langages, consistant essentiellement en des opérations booléennes sur les propositions. On peut, grâce à elle, étudier dans un cadre formel la valeur de vérité de formules relativement peu expressives. Insuffisante pour représenter des procédés de langage effectivement utilisés en informatique, linguistique ou en mathématiques, elle sert néanmoins de base à la construction de systèmes formels plus expressifs. La formalisation du raisonnement suivant lui échappe :

– *Certains étudiants assistent à tous les cours.*
– *Aucun étudiant n'assiste à un cours inintéressant.*

Peut-on conclure que tous les cours sont intéressants ?

En logique propositionnelle, on peut réaliser la déduction suivante :

$$\frac{\text{Le cours d'IA est intéressant,} \quad \text{Un cours intéressant attire de nombreux étudiants}}{\text{Le cours d'IA attire de nombreux étudiants}}$$

On peut facilement effectuer le même raisonnement avec une autre proposition : « le cours de sociologie est intéressant ». On pourrait envisager des choses intéressantes dans un autre contexte. Par exemple en remplaçant « cours de quelque chose » par « musée de quelque chose » et « étudiants » par « touriste », cela fait toujours du sens. Mais cela nécessite d'écrire une nouvelle proposition à chaque fois.

En fait, on aimerait dissocier « intéressant » de « cours » ou « musée », de telle sorte que ce soit une propriété du cours qui n'a pas nécessairement la même valeur selon que le cours soit de l'IA ou de la sociologie. On voudrait aussi pouvoir exprimer des relations entre plusieurs objets, par exemple que la salle du cours de sociologie est en face de la salle de TD d'IA. Enfin, on souhaiterait exprimer que des propriétés sont vraies pour « certains étudiant » ou pour « tous les cours ».

Le **logique des prédicats**, ou **logique du premier ordre** est par nature plus expressive que la logique des propositions, et permet de représenter ces types de connaissances relatifs à des environnements complexes. Elle est construite à partir de la logique propositionnelle et s'inspire du langage naturel pour définir :

– des **objets**, par exemple « IA », « sociologie », « Julien »
– des **fonctions** sur ces objets, « TD d'IA », « cours de sociologie »
– des **relations** entre ces objets, « Julien assiste au cours d'IA »

Tout comme en logique propositionnelle, on s'intéresse à la validité des relations existant entre les objets.

2.1 Structure

Pour écrire un énoncé en logique du premier ordre, nous allons utiliser un ensemble de symboles plus riche qu'en logique des propositions. On se donne :

– un ensemble (dénombrable) de **constantes** $\{a, b, c, \ldots\}$;
– un ensemble (dénombrable) de **variables** $\{x, y, z, \ldots\}$;
– un ensemble (dénombrable) de **fonctions** $\{f, g, h, \ldots\}$;
– un ensemble (dénombrable) de **prédicats**, ou **relations** $\{P, Q, \ldots\}$;
– des **connecteurs** logiques, $\{\neg, \wedge, \vee, \rightarrow, \ldots\}$, ainsi que les parenthèses '(' et ')' ;
– les **quantificateurs** universel $\forall$ et existentiel $\exists$ (voir plus loin).

2.1.1 Termes

Un **terme** est une expression logique qui renvoie à un objet. Les constantes, comme « Julien » ou « sociologie », ainsi que les variables, sont des termes. Un **terme composé** est construit à l'aide d'une fonction, par exemple « père(Julien) » ou « cours(IA) ».

2.1.2 Formules

Une **formule** en logique des prédicats se construit similairement à une formule en logique des propositions. En fait un prédicat va jouer un rôle analogue à une proposition. On doit en plus prendre en compte les quantifications :

1. $P(x_1, \ldots, x_n)$ est une formule atomique ;
2. $t_1 = t_2$ est une formule atomique[4] ;
3. si F est une formule, alors $\neg F$ est une formule ;
4. si F et G sont des formules, alors $(F \wedge G)$, $(F \vee G)$, $(F \rightarrow G)$, etc. sont des formules ;
5. si F est une formule et x une variable, alors $\forall x.F$ et $\exists x.F$ sont des formules.

2.1.3 Quantificateurs

Le quantificateur **universel** exprime le fait que tous les éléments d'un ensemble d'objets sur lequel s'exprime un prédicat vérifient ce prédicat, c'est-

[4]L'opérateur d'égalité permet de comparer deux termes.

à-dire $\forall x.P(x)$ est vrai revient à considérer que $P(a_1) \wedge \cdots \wedge P(a_n)$ est vrai, si $\{a_1, \ldots, a_n\}$ est le domaine de x[5].

De la même façon, le quantificateur **existentiel** exprime le fait qu'au moins un des éléments d'un ensemble d'objets sur lequel s'exprime un prédicat vérifie ce prédicat, c'est-à-dire $\exists x.P(x)$ est vrai revient à considérer que $P(a_1) \vee \cdots \vee P(a_n)$ est vrai, si $\{a_1, \ldots, a_n\}$ est le domaine de x.

Quantificateurs imbriqués Notons que l'ordre des quantificateurs n'est pas anodin. En effet, « tout le monde aime quelqu'un » s'écrirait $\forall x\ .\ (\exists y\ .\ Aime(x, y))$, qui n'a pas exactement le même sens que « il y a quelqu'un qui est aimé par tout le monde » qui s'écrirait $\exists y\ .\ (\forall x\ .\ Aime(x, y))$.

Liens entre $\forall$ et $\exists$ On a les lois de Morgan pour les quantificateurs :

$$\begin{aligned}
\neg\forall x.F &\equiv \exists x.\neg F\\
\neg\exists x.F &\equiv \forall x.\neg F\\
\forall x.F &\equiv \neg\exists x.\neg F\\
\exists x.F &\equiv \neg\forall x.\neg F
\end{aligned}$$

Illustrations :

– « Tout le monde déteste les brocolis » revient au même que « Il n'existe personne qui aime les brocolis » :

$$\forall x.\neg Aime(x, brocolis) \equiv \neg\exists x.Aime(x, brocolis)$$

– « Tout le monde aime les glaces » et « Il n'y a personne qui n'aime pas les glaces » sont équivalents :

$$\forall x.Aime(x, glaces) \equiv \neg\exists x.\neg Aime(x, glaces)$$

2.1.4 Exemples

Les énoncés que nous évoquions au début de cette section peuvent s'exprimer avec des formules de la logique du premier ordre.

1. « Certains étudiants assistent à tous les cours. »

$$\exists x.(Etudiant(x) \wedge (\forall y.Assiste(x, y)))$$

2. « Aucun étudiant n'assiste à un cours inintéressant. »

$$\neg\exists x.(Etudiant(x) \rightarrow (Assiste(x, y) \wedge \neg Interessant(y)))$$

Dans la seconde formule, on constate que la variable y n'est pas quantifiée : un telle variable est dite **libre**. Une variable quantifiée est dite **liée**.

[5] L'ensemble des valeurs que peut prendre x.

Exercice Exprimer les énoncés suivants en logique du premier ordre.

– « Tous les lions sont féroces. »

$$\forall x.(Lion(x) \rightarrow Feroce(x))$$

– « Quelques lions ne boivent pas de café. »

$$\exists x.(Lion(x) \wedge \neg Cafe(x))$$

– « Aucun singe n'est soldat. »

$$\forall x.(Singe(x) \rightarrow \neg Soldat(x))$$

– « Tous les singes sont malicieux. »

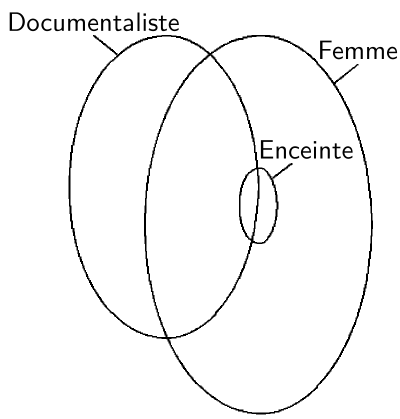
$$\forall x.(Singe(x) \rightarrow Malicieux(x))$$

Remarque On traduit couramment certaines expressions en logique du premier ordre :

– « Tous les A sont B. » : $\forall x.(A(x) \rightarrow B(x))$
– « Seuls les A sont B. » : $\forall x.(B(x) \rightarrow A(x))$
– « Aucun A n'est B. » : $\forall x.(A(x) \rightarrow \neg B(x))$
– « Quelques A sont B. » : $\exists x.(A(x) \wedge B(x))$

2.2 Logique et théorie des ensembles

La logique du premier ordre entretient un rapport très étroit avec la théorie des ensembles. On peut en effet voir un ensemble comme une collection d'objets satisfaisant une certaine propriété. Ainsi, on pourra exprimer les caractéristiques d'un individu par une formule de la logique des prédicats. Par exemple, $\exists x.(Femme(x) \wedge Documentaliste(x))$ correspondra à l'intersection des ensembles Femme et Documentaliste. Le formule $\forall x.(Enceinte(x) \rightarrow Femme(x))$ exprimera le fait que l'ensemble Enceinte est inclus dans l'ensemble Femme. Et ainsi de suite.



Références

[1] R. Lassaigne et M. de Rougemont. *Logique et fondements de l'informatique*. Hermes, 1993.

[2] M.-D. Popelard et D. Vernant. *Éléments de logique*. Mémo Seuil, 1998.

[3] S. Russel et P. Norvig. *Intelligence artificielle* (2e édition). Pearson Education, 2006.