Programmation Logique



Licence 3 Génie Logiciel

Année Académique: 2025-2026

ECOLE DES MINES, DE L'INDUSTRIE ET DE LA GEOLOGIE

Alio Boubacar Goga, Researcher

Plan du cours

Plan du cours

Section 1 : Logique des propositions

Section 2 : : Logique des prédicats

Section 3 : Introduction à la programmation logique : Prolog

Section 4 : Application de la logique dans l'Intelligence Artificielle (IA)

Organisation du cours

- **C**ours **M**agistral (**CM**) : 06 H
- **T**ravaux **D**irigés (**TD**) : 06 H
- **T**ravaux **P**ratiques (**TP**) : 03 H

gogaalioboubacar@gmail.com

Section 1 : Logique des propositions

❖ Introduction et définitions:

❑ La logique est l'étude des lois du raisonnement.

❑ La logique des propositions ou calcul propositionnel étudie les formes de raisonnement dont seules les valeurs de vérité des énoncés déterminent en dernière instance la validité du dit raisonnement.

❑ La proposition est tout énoncé dont on peut décider s'il est vrai ou faux sans avoir à en analyser.

❑ Exemple 1 : l'énoncé « il chante » est une proposition qui peut avoir la valeur de vérité vraie ou fausse.

❑ Cet énoncé ci-dessus peut être considéré comme élémentaire.

❑ On peut former, des énoncés plus complexes en reliant les énoncés élémentaires par des connecteurs logiques.

Section 1 : Logique des propositions

❑ On nommera ces énoncés obtenus formules qui elles aussi seront susceptibles d'être vraies ou fausses.

❑ Exemple 2 :

- Moussa mange ou Ali parle = P∪Q
- Ali mange et Moussa chante = R∩S
- Si Ali mange alors Moussa parle = R→ Z
- Je ne mange pas = ¬ Po

❑ Ou, et, si … alors, ne … pas sont des connecteurs logiques.

❑ Dans cette section, nous étudierons ces types d'énoncés.

❖ Le vocabulaire de la logique des propositions:

L'alphabet du langage de la logique des propositions est défini par :

❑ Un ensemble de symboles appelés variables propositionnelles notés P, Q, R, S, …

❑ Deux constantes logiques : ⊤ (Vrai) et ⊥ (faux).

❑ Des connecteurs :

Section 1 : Logique des propositions

- ¬ : "**non** " pour la négation
- ∩ : "**et** « pour la conjonction
- ∪ : "**ou** « pour la disjonction
- → : "**implique** " pour l'implication
- ↔ : "**équivalent** " pour l'équivalence
- "( )" : les parenthèses.

❑ Les variables propositionnelles sont qualifiées de formules atomiques ainsi que ⊤ (Vrai) et ⊥ (faux).

❑ ¬ est un connecteur unaire alors que les autres (∩, ∪ , → , ↔) sont des connecteurs binaires.

❖ Les formules (ou syntaxes): L'ensemble des formules de la logique propositionnelle est l'ensemble F tel que :

❑ toute variable propositionnelle est l'élément de F.

❑ ⊤ (Vrai) et ⊥ (faux) sont éléments de F.

❑ si A ∈ F alors (¬ A) ∈ F.

Section 1 : Logique des propositions

- si A ∈ F et B ∈ F alors (A∩ B), (A∪B), (A→B), (A↔B) sont éléments de F.
- A et B sont soit des formules atomiques, soit des formules complexes correctement formés.
- Exemple :

((P→Q) ∩ R) ∪ (¬ (P∩Q))

((P→ R) → Q)

(P↔R) ∩ (Q∪R)

Sont-ils valides ?

¬ → P

¬→ (P) ↔)R

P∪

P¬(⊥)

Sont-ils valides ?

- Ordre de précédence entre les connecteurs :

Section 1 : Logique des propositions

- ¬ est prioritaire sur ∩, ∪ , → , ↔
- ¬P∪R doit se lire ((¬P)∪R)
- ¬P → Q doit se lire ((¬P) → Q)
- ∩ et ∪ sont prioritaires sur → et ↔
- P→ Q∪S doit se lire (P→ (Q∪S))
- P∩Q→ R∪P doit se lire ((P∩Q)→ (R∪P))
- les parenthèses permettent de contre-carrer ces règles de priorité. Elles permettent aussi de rendre une formule plus lisible.

❑ Tout élément de F(A) est une sous formule de A.

Exemple 3 : A=((P→Q)∩(¬(Q→R)))

F(A)={P ;Q ;R ;P→Q ; Q→R ; ¬(Q→R}

Section 1 : Logique des propositions

❖ Les connecteurs et leur interprétation:

Rappelons que les propositions (ou formules) sont des énoncés susceptibles d'avoir des valeurs de vérité vraie ou fausse.

❑ P est vraie lorsque ¬P est fausse, et fausse sinon.

❑ P∩Q est vraie lorsque P et Q sont toutes les deux (2) vraies et fausse dans tous les cas.

❑ P∪Q est vraie lorsque l'une de ces propositions est vraie, mais aussi lorsqu'elles sont toutes les deux (2) vraies, et fausse lorsqu'elles sont toutes les deux (2) fausses.

❑ P→Q est fausse lorsque P est vraie et Q fausse ; mais vraie dans les autres cas.

❑ P↔Q est vraie lorsque P et Q ont la même valeur de vérité, et fausse dans le cas contraire.

❖ Fonction d'interprétation

Etant donné un ensemble de proposition P, une fonction d'interprétation ou valuation V est une application du type :

Section 1 : Logique des propositions

❑ V : P→ {Vrai, Faux}. C'est-à-dire une fonction qui assigne à chaque proposition de P la valeur Vraie ou fausse.

❑ on suppose connu une restriction aux formules qui sont des variables propositionnelles ; telle que :

- V(⊥) = faux ; V(⊤) = Vrai
- V(¬A) = Vrai ssi V(A)=faux
- V(A∩B) = Vrai ssi V(A)=V(B)
- V(A→B) = Faux ssi V(A)=Vrai et V(B) = Faux
- V(A↔B) = Vrai ssi V(A)=V(B).
- pour la valuation d'une formule complexe, il suffit d'associer aux différentes variables propositionnelles, des valeurs de vérité et d'entendre par induction sur toute la formule.

▪ Cela se visualise aisément à l'aide des tables de vérité de la négation ; de la conjonction, de la disjonction, de l'implication et de l'équivalence.

Section 1 : Logique des propositions

Tables de vérité :

A	¬A
Faux	Vrai
Vrai	Faux

A	B	A∪B
Faux	Faux	Faux
Vrai	Faux	Vrai
Faux	Vrai	Vrai
Vrai	Vrai	Vrai

A	B	A→B
Faux	Faux	Vrai
Vrai	Faux	Faux
Faux	Vrai	Vrai
Vrai	Vrai	Vrai

A	B	A∩B
0	0	0
1	0	0
0	1	0
1	1	1

A	B	A↔B
Faux	Faux	Vrai
Vrai	Faux	Faux
Faux	Vrai	Faux
Vrai	Vrai	Vrai

Section 1 : Logique des propositions

- nous notons V⊨A ssi v(A) = Vrai, qui se dit « V satisfait A»
- exemple : étudier la valuation de A = P∩(¬Q∪¬P)

P	Q	¬P	¬Q	¬Q∪¬P	P∩(¬Q∪¬P)
0	0	1	1	1	0
1	0	0	1	1	1
0	1	1	0	1	0
1	1	0	0	0	0

Section 1 : Logique des propositions

❑ Pour une formule donnée, trois (3) cas peuvent se présenter :

○ pour toute valuation V, la formule est vraie (c'est-à-dire il n'y a que des 1 (vrai) dans la dernière colonne de sa table de vérité.

○ pour toute valuation V, la formule est fausse (c'est-à-dire il n y a que des 0 (faux) dans la dernière colonne de sa table de vérité.

○ il existe une valuation V qui donne la valeur vraie et une autre qui donne la valeur fausse (il y a au moins un 1 et au moins un 0 dans la dernière colonne de sa table de vérité).

Remarque :

○ la première catégorie de formule regroupe les tautologies (valides)

○ la deuxième catégorie de formule regroupe les contradictions

○ la troisième catégorie de formule regroupe les formules satisfaisables (ni tautologie, ni contradiction)

- une formule est satisfaisable s'il existe au moins une valuation pour laquelle la formule est vraie. Donc la 1ère et la 3ème catégories regroupent des formules satisfaisables.
- une conséquence directe de ces définitions est que, une formule propositionnelle est une tautologie si sa négation n'est pas satisfaisable.

❖ Equivalence sémantique :

❑ deux formules A et B sont sémantiquement équivalent (A≡B) ssi pour toute valuation V on a V(A)=V(B).

Section 2 : Logique des prédicats

❖On a vu que la logique des propositions s'occupe des énoncés dont la signification est d'être vraie ou fausse.

❑ exemple :

○ Anne chante

○Si Jean marche alors Anne chante

❖ Cependant, elle est restreint à un calcul mettant en jeu des propositions dont la particularité est d'être indécomposable.

❑ considérons par exemple les énoncés suivants :

○ dans toute salle d'examen, il y a un étudiant qui, s'il échoue alors tout le monde échoue.

○ certains étudiants assistent à tous les cours.

○ il existe au moins un entier relatif tel que son carré soit égale à 9.

○ ces énoncés ci-dessus peuvent-ils être formalisés en logique propositionnelle.

Section 2 : Logique des prédicats

❖ Ce qui manque à la logique des propositions c'est de pouvoir :

❑ prendre en compte des objets, éléments d'un ensemble donné :un entier n en générale, entier 9 en particulier.

❑ construire d'autres objets par intermédiaire d'opérateurs : calculer le carré d'un entier relatif.

❑ rendre compte de certains propriétés vérifiées par les objets : être intéressant ou non, être positif ou non, …

❑ indiquer qu'éventuellement tous les objets, éléments d'un ensemble donné, vérifient une certaine propriété : tous cours sont intéressants ; il existe des cours intéressants.

❖ La logique des prédicats ou logique du premier ordre : elle nous permettra de modifier ces situations.

❑ par exemple l'énoncé « tous les cours sont intéressants » peut être modélisé par $\forall$ x .cours (x)→ intéressant (x).

Section 2 : Logique des prédicats

❑ le terme prédicat est utilisé pour :

○ représenter une propriété du type « a est p » comme dans l'énoncé « Socrate est mortel »

○ ou représenter une propriété du type « a, b, c,… ont entre eux une relation R comme dans l'énoncé » « Moussa, David, Sani sont des frères ».

○ ici les a, b, c jouent le rôle de sujet tandis que p et R jouent le rôle de prédicats.

○ le fait d'affirmer que tous les objets d'un ensemble donné vérifient une certaine propriété ou qu'il existe (au moins) un objet de cet exemple qui vérifie telle propriété s'appelle, la quantification. Elle sera représentée par les symboles ∀ (pour tout) et ∃ (il existe) dits respectivement quantificateur universel et quantificateur existentiel. Ces quantificateurs seront suivis du nom de la variable sur laquelle porte la quantification puis du prédicat concerné.

▪ « il existe au moins un entier relatif tel que son carré soit égale à 4 » : ∃ x. (f(x)=4).

Section 3 : Introduction à la programmation logique : Prolog

❖ Prolog est un langage de programmation logique, c'est-à-dire le programme ne se représentera pas sous forme d'une suite d'instructions comme dans la programmation impérative.

❖ Prolog est un langage de programmation issu de la logique computationnelle.

- ❑ son nom vient de « **PROgrammation LOGique** »
- ❑ son fondateur est Alain Colmerauer de l'université de Marseille
- ❑ Ce langage a été créé 1973.
- ❑ Comme langage de programmation, Prolog utilise un formalisme différent des autres langages pour l'écriture des programmes.
- ❑ Prolog permet d'utiliser la logique pour traiter des informations avec un ordinateur, c'est-à-dire d'utiliser la logique comme un langage de programmation.
- ❑ Prolog est fondé sur la logique mathématique afin de stocker et traiter des informations sous forme symbolique.

- ces informations symboliques sont issues de la modélisation des connaissances que possède un être intelligent.

❖ Un programme prolog est constitué d'une suite de prédicats ou clauses de deux (2) types :

- les faits : elles décident les différentes relations entre certains objets. Ce sont des données élémentaires que l'on considère vraies des formules atomiques.

○pour répondre aux requêtes de l'utilisateur, prolog cherche des preuves dans les faits.

○ un fait se termine toujours par un point (.).

exemple :

pere (Moussa, Issa).

astre (terre).

capitale (Niamey, Niger).

❑ les règles : un programme prolog peut contenir des règles. Elles sont des relations qui permettent à partir des faits (hypothèses) d'établir de nouveaux faits par induction.

exemple :

la relation si x est un garçon alors x aime les ballons se traduit par la règle : aime (X, ballon) : - garçon

❑ les constantes en prolog sont des chaînes de caractère (minuscule).

❑ les variables sont des chaines de caractères (majuscule).

❑ le «si » s'écrit « :- » en prolog et correspond à ← .

garçons (X)→ aime (X, ballon)≡ aime (X, ballon) :-garçon (X).

❑ le Et logique (conjonction) correspond à une virgule en prolog.

❑ le Ou logique (disjonction) correspond à un point virgule.

○si on est le père du père de quelqu'un alors on est son grand-père. Cela se traduit par :

❑ grandpère (X,Y) :-père (X,Z), père (Z,Y).

❑ Une clause est une formule (universellement quantifiée) dont le corps est une disjonction de formules anatomiques ou la négation des formules atomiques.

exemple :

$A(x) \cap B(y)$ n'est pas une clause

$\forall x.\forall y .(A(x) \cup B(y))$ est une clause

$\forall x.\forall y .(\neg A(x) \cup B(y))$ est une clause.

En résumé :

La programmation en Prolog consiste en :

- ❑ la spécification des faits concernant les objets,
- ❑ la construction des règles concernant les relations entre objets,
- ❑ la réponse à des questions posées concernant l'existence des objets et/ou les relations entre objets.

NB: Les faits, règles et questions sont des expressions logiques.

Section 4 : Application de la logique dans l'Intelligence Artificielle

❖ La logique a été utilisée dans l'Intelligence Artificielle (IA) à plusieurs endroits :

❑ utiliser des expressions logiques pour représenter des faits, des connaissances

❑ utiliser la logique pour inférer des conclusions

❖ Dans une application en IA, face à un problème, le premier travail consiste à d'abord créer une représentation pour ce problème.

❑ cette étape est appelée la conceptualisation.

❑ la logique a été d'abord utilisée pour représenter des faits qu'on observe dans une application.

○ par exemple, dans un monde de blocks, si on observe que le block a est posé sur le block b, alors on peut représenter ce fait par l'expression sur (a, b).

Mini Projet : créer un système expert en choisissant un domaine d'application.