

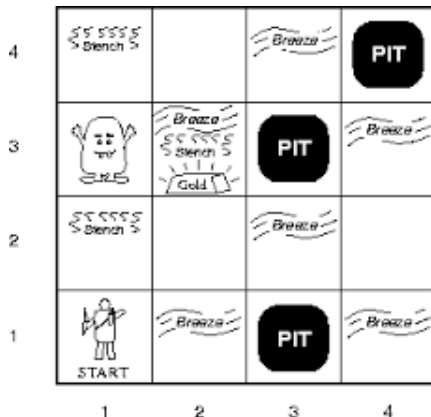
Agents logiques

Rym Guibadj

LISIC, EILCO

Un premier exemple : le monde du Wumpus

- Environnement
 - Grille 4x4
 - Depart en [1,1]
- Effecteurs
 - Pivoter droite
 - Pivoter gauche
 - Avancer
 - Saisir
 - Tirer
- Capteur
 - Odeur (Wumpus)
 - Brise (Puits)
 - Lueur (or)
 - Choc (mur)
 - Cri (Mort)
- But
 - Trouver l'or et sortir



Un premier exemple

Propriété

Si l'agent tire une conclusion en s'appuyant sur les informations disponibles, alors cette conclusion est garantie correcte si les informations sont elles-mêmes correctes

Suite du cours

Comment construire un tel agent ?

Logique propositionnelle

- Formules propositionnelles
 - Définies à l'aide de **constantes**, de **variables** et de **connecteurs**
- Constantes
 - **vrai** ou **faux**
- Variables
 - Ensemble dénombrable
 - Représentées par des lettres : **p, q, r, s ...**
- Connecteurs
 - Unaire : \neg not
 - Binaire : \wedge (et), \vee (ou), \Rightarrow (implique), \Leftrightarrow (ssi)
 - Priorité : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ (parenthèses possibles)
- Exemple :
 - $p \wedge q \vee \neg r \Rightarrow \text{Vrai}$

Sémantique de la LP

- But de la sémantique
 - Associer une signification aux formules
 - Définir la valeur de vérité (vrai ou faux)
- Table de vérité

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
faux	faux					
faux	vrai					
vrai	faux					
vrai	vrai					

Sémantique de la LP

- But de la sémantique
 - Associer une signification aux formules
 - Définir la valeur de vérité (vrai ou faux)
- Table de vérité

P	Q	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \Rightarrow Q$	$P \Leftrightarrow Q$
faux	faux	vrai	faux	faux	vrai	vrai
faux	vrai	vrai	faux	vrai	vrai	faux
vrai	faux	faux	faux	vrai	faux	faux
vrai	vrai	faux	vrai	vrai	vrai	vrai

Sémantique de la LP

Validité

Une proposition est valide si elle est vraie pour toutes les assignations

Satisfiabilité

Une proposition est satisfiable si elle est vraie pour au moins une assignation

Exemple

Soit la formule : $F = (\neg p \Rightarrow q) \wedge (p \Rightarrow r)$

F est valide ? F est satisfiable ?

Sémantique de la LP

Exemple

Soit la formule : $F = (\neg p \Rightarrow q) \wedge (p \Rightarrow r)$

F est valide ? F est satisfiable ?

p	q	r	$\neg p$	$\neg p \Rightarrow q$	$p \Rightarrow r$	$(\neg p \Rightarrow q) \wedge (p \Rightarrow r)$
faux	faux	faux	vrai	faux	vrai	faux
faux	faux	vrai	vrai	faux	vrai	faux
faux	vrai	faux	vrai	vrai	vrai	vrai
faux	vrai	vrai	vrai	vrai	vrai	vrai
vrai	faux	faux	faux	vrai	faux	faux
vrai	faux	vrai	faux	vrai	vrai	vrai
vrai	vrai	faux	faux	vrai	faux	faux
vrai	vrai	vrai	faux	vrai	vrai	vrai

\Rightarrow

F est satisfiable mais non valide

Conséquence logique

Définition

- Soit $E = F_1, \dots, F_n$, un ensemble non vide de formules et G une formule
- G est une conséquence logique de E ($F_1, \dots, F_n \models G$) ssi toutes les interprétations pour lesquelles F_1, \dots, F_n sont vraies, G est vraie

Observations

- $F_1, \dots, F_n \models G$ ssi $(F_1 \wedge \dots \wedge F_n) \Rightarrow G$ est valide
- si E est insatisfiable alors $F_1, \dots, F_n \models G$
- si G est valide, elle est C. L. de n'importe quel ensemble de formule
- $E \models G$ ssi $E \cup \neg G$ est insatisfiable

Exemple : conséquences logiques de F ?

- $F = (p \Rightarrow q) \wedge (p \vee r)$
- $G = q \vee r$
- $H = p \Rightarrow r$

Exemple : conséquences logiques de F ?

- $F = (p \Rightarrow q) \wedge (p \vee r)$
- $G = q \vee r$
- $H = p \Rightarrow r$

p	q	r	F	G	H
faux	faux	faux	faux	faux	vrai
faux	faux	vrai	vrai	vrai	vrai
faux	vrai	faux	faux	vrai	vrai
faux	vrai	vrai	vrai	vrai	vrai
vrai	faux	faux	faux	faux	faux
vrai	faux	vrai	faux	vrai	vrai
vrai	vrai	faux	vrai	vrai	faux
vrai	vrai	vrai	vrai	vrai	vrai

- $F \models G$
- H n'est pas conclusion logique de F

Conséquence logique

Notion

La notion de conséquence logique permet de vérifier certains raisonnements formulés en langage naturel

Exemple

”Si je vous paie mon installation TV c’est qu’elle marche. Or, elle ne marche pas, donc, je ne vous paierai pas”

Conséquence logique

Notion

La notion de conséquence logique permet de vérifier certains raisonnements formulés en langage naturel

Exemple

"Si je vous paie mon installation TV c'est qu'elle marche. Or, elle ne marche pas, donc, je ne vous paierai pas"

On formalisme :

- p : "je vous paierai votre installation TV"
- m : "mon installation marche"

On écrit le raisonnement : $\{p \Rightarrow m, \neg m\} \models \neg p$

$p \Rightarrow m$ et $\neg m$ sont les prémisses et $\neg p$ la conclusion

Récapitulatif

La logique est un langage formel permettant de représenter l'information de sorte qu'on puisse en déduire des conclusions, et donc de nouvelles informations

Syntaxe

définit l'alphabet et la structure des phrases du langage

Sémantique

définit le sens des phrases, la vérité d'un énoncé

D'un point de vue d'un agent en IA

Un agent dispose d'une base de connaissance et d'un moyen de faire des déductions pour choisir ses actions

Base de connaissances

Un ensemble de faits ou de formules vraies

Méthodes de preuves

- Vérification de modèles
 - Enumération (par table de vérité)
 - Par réfutation
 - Algorithme de résolution
- Application de règles d'inférence.
 - Génération de nouvelles formules à partir de formules vraies.
 - Preuve : séquence d'application de règles d'inférence

Démonstration par réfutation

Principe

- Une démonstration par l'absurde
- Pour montrer $KB \models q$ on peut montrer que : $KB \cup \neg q \Rightarrow \emptyset$
- Montrer que $KB \wedge \neg q$ n'est pas valide

Remarques

- Il faut vérifier toutes les interprétations possibles donc 2^n cas si on a n variables propositionnelles
- Algorithme de complexité exponentiel \Rightarrow en pratique peu utilisable

Algorithme de résolution

Règle de résolution

A partir de $(p_1 \vee \dots \vee p_m), (q_1 \vee \dots \vee q_n)$, si $p_i = \neg q_j$, on déduit :
 $(p_1 \vee \dots \vee p_{i-1} \vee p_{i+1} \vee \dots \vee p_m \vee q_1 \vee \dots \vee q_{j-1} \vee q_{j+1} \vee \dots \vee q_n)$

Principe

Pour montrer $\{F_1, \dots, F_n\} \models G$, il faut :

- Mettre la formule $F_1 \wedge \dots \wedge F_n \wedge \neg G$ sous forme normale conjonctive.
- Appliquer la règle de la résolution sur les clauses jusqu'à ce que on trouve la clause vide ϕ

Règles d'inférence

Principe

- Appliquer un schéma de règles qui conduisent au but recherché
- $F_1, \dots, F_n / p$: à partir de F_1, \dots, F_n on peut montrer p

Quelques règles d'inférence

$p \Rightarrow q, p / q$	(le modus ponens)
$p \wedge q / q$	(élimination de la conjonction)
$p \Leftrightarrow q / (p \Rightarrow q) \wedge (q \Rightarrow p)$	(élimination de l'équivalence)
$(p \Rightarrow q) \wedge (q \Rightarrow p) / p \Leftrightarrow q$	(élimination de la double implication)

Introduction à la déduction

• Notations

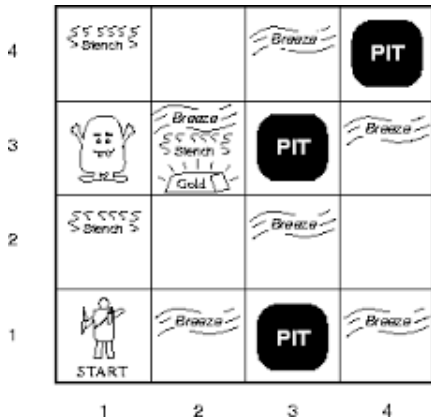
- $p_{i,j}$ est vrai s'il y a un puits en $[i, j]$
- $b_{i,j}$ est vrai s'il y a une brise en $[i, j]$

• Connaissance

- $F_1 : \neg p_{1,1}$
- $F_2 : \neg b_{1,1}$
- $F_3 : b_{1,2}$
- $F_4 : \neg p_{1,2}$
- $F_5 : b_{1,1} \Leftrightarrow (p_{1,2} \vee p_{2,1})$
- $F_6 : b_{1,2} \Leftrightarrow (p_{1,1} \vee p_{2,2} \vee p_{1,3})$

• Objectif

- On veut savoir s'il y a un puits en $[2, 1]$, $[2, 2]$
- On cherche la vérité de $\neg p_{2,1}, \neg p_{2,2}$



La logique du premier ordre

Motivation

Pour résoudre des problèmes complexes qui demandent des connaissances d'un expert, nous avons besoin d'un langage permettant de :

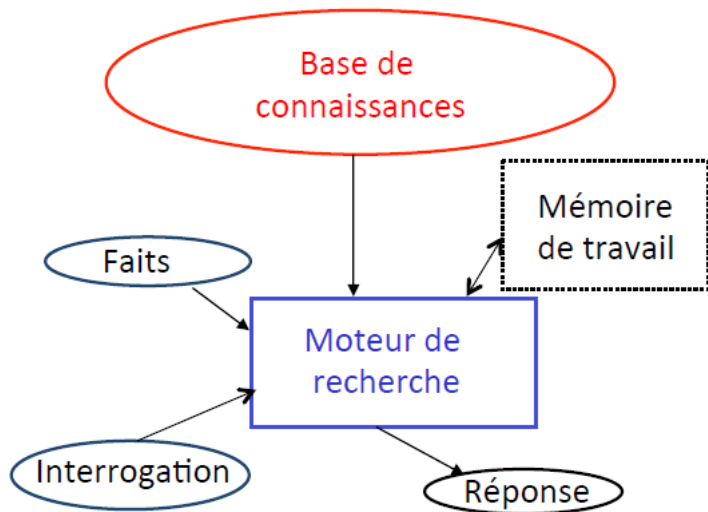
- représenter les **connaissances d'un expert** facilement.
- faire des **déductions logiques** avec ces connaissances.

La logique du premier ordre

Exemples de raisonnement déductif :

- Dédire la maladie du patient et le traitement approprié, à partir de :
 - Symptômes d'un patient.
 - Règles de causalité entre les symptômes et les pathologies.
 - Règles de causalité sur les traitements.
- Diagnostiquer le problème d'un véhicule à partir de :
 - Symptômes d'un véhicule.
 - Règles de causalité pour la mécanique auto.

Schéma d'un système expert



Introduction

Peut on modéliser les situations suivantes en logique propositionnelle :

- certains étudiants assistent à tous les cours
- aucun étudiant n'assiste à un cours non intéressant
- dans toute salle d'examen, il y a un étudiant qui, s'il échoue, alors tout le monde échoue

Avec la logique des propositions, on peut

- représenter l'information,
- déduire / inférer de nouvelles connaissances.

Mais on ne peut pas

- généraliser
- utiliser des relations

Introduction

La logique des propositions

suppose que le monde contient des **faits**

La logique des prédicats

suppose (comme le langage courant) que le monde contient :

- **Des objets** : Gens, maisons, nombres, ...
- **Des relations** : Frères, plus grand que, ...
- **Des fonctions** : Le père de, le meilleur ami de, la deuxième mi-temps de, ...

Syntaxe

- Constantes
 - Michel, 2, ULCO, ...
- Relations (prédicats)
 - vrai, $>$, ...
- Fonctions
 - $\sqrt{\dots}$, père, ...
- Variables
 - x, y , ...
- Connecteurs
 - \Rightarrow , \Leftrightarrow , \vee , \wedge , \neg
- Égalité (entre termes)
 - $=$
- Quantificateurs
 - \forall , \exists

Formules atomiques

Formule atomique

prédicat($terme_1, \dots, terme_n$)

Termes

- Constante
- Variable
- Fonction($terme_1, \dots, terme_n$)

Exemples

- Frère(Paul, Etienne)
- Marié(père(Michel), mère(Sophie))

Formules complexes

Définition

- Construites à partir des formules atomiques
- Ajouts de connecteurs

Exemple

- $\text{Frères}(\text{John}, \text{Richard}) \Rightarrow \text{Frères}(\text{Richard}, \text{John})$

Quantification universelle

- \forall <variables> <formule>
- “Tous les étudiants de Calais aiment l’intelligence artificielle” :
- $\forall x \text{ Etudiant}(x, \text{Calais}) \Rightarrow \text{Aime}(x, \text{IA})$.
- $\forall x P$ ssi P est vrai avec x pouvant être tous les objets du modèle.
- Réécriture possible avec un ensemble de conjonction
 $(\text{Etudiant}(\text{Albert}, \text{Calais}) \Rightarrow \text{Aime}(\text{Albert}, \text{IA}) \wedge$
 $(\text{Etudiant}(\text{Bertrand}, \text{Calais}) \Rightarrow \text{Aime}(\text{Bertrand}, \text{IA}) \wedge$
 ...

Quantification existentielle

- \exists <variables> <formule>
- “Il existe au moins un étudiant de Calais qui aime l'intelligence artificielle” :
- $\exists x \text{ Etudiant}(x, \text{Calais}) \wedge \text{Aime}(x, \text{IA})$.
- $\exists x P$ ssi P est vrai avec x pouvant être (au moins) un objet du modèle.
- Réécriture possible avec un ensemble de disjonction
 $(\text{Etudiant}(\text{Albert}, \text{Calais}) \wedge \text{Aime}(\text{Albert}, \text{IA})) \vee$
 $(\text{Etudiant}(\text{Bertrand}, \text{Calais}) \wedge \text{Aime}(\text{Bertrand}, \text{IA})) \vee$
 ...

Attention

- \Rightarrow est le connecteur principal avec \forall
 - Une erreur courante est d'utiliser \wedge
 - $\forall x \text{ Etudiant}(x, \text{Calais}) \wedge \text{intelligent}(x)$ signifie :
 - "tous les étudiants sont à Calais et tous les étudiants sont intelligents"
- \wedge est le connecteur principal avec \exists
 - Une erreur courante est d'utiliser \Rightarrow
 - $\exists x \text{ Etudiant}(x, \text{Calais}) \Rightarrow \text{intelligent}(x)$ est vrai :
 - s'il existe quelqu'un qui n'est pas étudiant à Calais

Propriétés des quantificateurs

Equivalence

- $\forall x \forall y$ est équivalent à $\forall y \forall x$
- $\exists x \exists y$ est équivalent à $\exists y \exists x$
- **Attention :** $\forall x \exists y$ n'est pas équivalent à $\exists y \forall x$
 - $\exists y \forall x \text{ Etudie}(x,y)$: "il existe un module étudié de tous"
 - $\forall x \exists y \text{ Etudie}(x,y)$: "tout le monde suit un module (pour tout étudiant, il existe un module au quel il est inscrit)"

Propriétés des quantificateurs

Equivalence

- $\forall x \forall y$ est équivalent à $\forall y \forall x$
- $\exists x \exists y$ est équivalent à $\exists y \exists x$
- Attention : $\forall x \exists y$ n'est pas équivalent à $\exists y \forall x$
 - $\exists y \forall x \text{ Etudie}(x,y)$: "il existe un module étudié de tous"
 - $\forall x \exists y \text{ Etudie}(x,y)$: "tout le monde suit un module (pour tout étudiant, il existe un module au quel il est inscrit)"

Dualité

- $\forall x \text{ Aime}(x, \text{les glaces})$ est équivalent à $\neg \exists x \neg \text{Aime}(x, \text{les glaces})$
- $\exists x \text{ Aime}(x, \text{les épinards})$ est équivalent à $\neg \forall x \neg \text{Aime}(x, \text{les épinards})$

Modélisation d'expressions courantes

Exemples

$C(x)$: x est un client , $F(x)$: x est français, $V(x)$: x est végétarien,
 $E(x)$: x est un employé

- Tous les clients sont français :
- Quelques clients ne sont pas végétariens :
- Aucun employé n'est un client :
- Tous les employés sont végétariens :

Modélisation d'expressions courantes

Exemples

$C(x)$: x est un client , $F(x)$: x est français, $V(x)$: x est végétarien,
 $E(x)$: x est un employé

- Tous les clients sont français : $\forall x, C(x) \rightarrow F(x)$
- Quelques clients ne sont pas végétariens : $\exists x, C(x) \wedge \neg V(x)$
- Aucun employé n'est un client : $\forall x, E(x) \rightarrow \neg C(x)$
- Tous les employés sont végétariens : $\forall x, E(x) \rightarrow V(x)$

Modélisation d'expressions courantes

Exemples

$C(x)$: x est un client , $F(x)$: x est français, $V(x)$: x est végétarien,
 $E(x)$: x est un employé

- Tous les clients sont français : $\forall x, C(x) \rightarrow F(x)$
- Quelques clients ne sont pas végétariens : $\exists x, C(x) \wedge \neg V(x)$
- Aucun employé n'est un client : $\forall x, E(x) \rightarrow \neg C(x)$
- Tous les employés sont végétariens : $\forall x, E(x) \rightarrow V(x)$

Notes

- 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 : $\exists x, A(x) \wedge B(x)$