INTRODUCTION

L’Intelligence Artificielle, IA , est définie comme « la construction de programmes informatiques qui s’adonnent à des tâches qui sont, pour l’instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau»

UNE DEFINITION DIFFICILE

Il existe différentes définitions de l'IA :

➢L'adjectif 'Artificielle': ce type d'intelligence est le résultat d'un processus crée par l'homme, plutôt que d'un processus naturel biologique

La notion d'intelligence est difficilement cernable :

➢La capacité d'acquérir et de retenir les connaissances, d'apprendre ou de comprendre grâce à l'expérience

➢L'utilisation de la faculté de raisonnement pour résoudre des problèmes, et pour répondre rapidement et de manière appropriée à une nouvelle situation

UNE DEFINITION DIFFICILE

En d'autres termes :

L'intelligence Artificielle est :

- ➢ Penser comme un humain
- ➢ Penser de manière rationnelle
  (processus de pensée et de réflexion)
- ➢ Agir comme un humain
- ➢ Agir de manière rationnelle
  (comportement et actions)

DOMAINES D’ APPLICATION DE L’ IA

Tout domaine où l’on est amené à résoudre des problèmes difficiles

- L’ingénierie(notamment construction des robots)
- Les sciences de la cognition pour la simulation de l’intelligence humaine
- Diagnostic médical
- Planification et ordonnancement des tâches
- Expertise et conseils
- Vision artificielle (permettre aux ordinateurs de comprendre les images et la vidéo ex : reconnaître des visages)
- Apprentissage automatique (concevoir des programmes qui peuvent s’auto-modifier en fonction de leur expérience)

Etc ….

POURQUOI L'IA ET PAS LA PROGRAMMATION PROCÉDURALE CLASSIQUE ?

Programmation procédurale	Intelligence Artificielle
exécute des instructions	utilise des connaissances et des règles de déduction
plus près du fonctionnement de la machine	plus près du fonctionnement de l'être humain
plus adapté aux traitements numériques	plus adaptée aux traitements symboliques
utilise beaucoup de calculs	utilise beaucoup d'inférences
suit des algorithmes rigides et exhaustifs	fait appel à des heuristiques et à des raisonnements incertains
n'est généralisable qu'à une classe de problèmes semblables	est généralisable à des domaines différents

SYSTÈMES FORMELS

Définition

Un système formel est un ensemble de données purement abstrait, sans lien avec l'extérieur, qui décrit les règles de manipulation d'un ensemble de symboles traités de façon uniquement syntaxique, c'est-à dire sans considération de sens (sémantique)

Il est constitué :

1°/ d'un alphabet fini de symboles;

2°/ d'un procédé de construction des mots du système formel;

3°/ d'un ensemble d'axiomes qui sont des mots;

4°/ d'un ensemble fini de règles de déduction qui permettent de déduire d'un ensemble fini de mots un autre ensemble de mots.

SYSTÈMES FORMELS

Définition

On appelle système formel $S = (\Sigma, F, A, R)$ la donnée de :

➢Alphabet Σ qui peut être un ensemble fini ou infini dénombrable, auquel cas on se donne une numérotation fixée une fois pour toute de Σ

➢F (ensemble des formules bien formées) : sous-ensemble récursif de l'ensemble de toutes les suites finies d'éléments de Σ i.e. Il existe un programme P qui, pour la donnée de f, indique au bout d'un temps fini si $f \in F$ ou $f \notin F$

SYSTÈMES FORMELS

Définition

- **A** (ensemble d'axiomes) : un sous-ensemble récursif de F

<u>i.e</u>. Il existe un programme Q qui, pour la donnée de f, indique au bout d'un temps fini si $f \in A$ ou $f \notin A$

- **R** : ensemble fini de règles de production, décidables, définies sur F

<u>i.e</u>. Pour chaque r_i, il existe un programme Q_i, qui pour la donnée de f_1, …, f_l,g indique (au bout d'un temps fini) si on peut déduire g à partir de f_1, ….. f_l par la règle r_i

SYSTÈMES FORMELS

Exemple

$\Sigma = \{1, +, =\}$

$F = \{1^n + 1^m = 1^p\}$

$A = \{1 + 1 = 11\}$

$R = \{r1, r2\}$

r1 : $1^n + 1^m = 1^p \vdash 1^{n+1} + 1^m = 1^{p+1}$

r2 : $1^n + 1^m = 1^p \vdash 1^n + 1^{m+1} = 1^{p+1}$

SYSTÈMES FORMELS

Exemple

11+111=11111 théorème?

F0: 1 + 1=11 (axiome)

h |— g et g |— n alors h |— n

F1: 1 + 11=111 (r2 sur f0)

F2: 11+11=1111(r1sur F1))

F3: 11+111=11111(r2sur f2)

1+1+1=111 théorème?

$1^n + 1^m = 1^{n+m}$

r1 : $1^n + 1^m = 1^p$ |--- $1^{n+1} + 1^m = 1^{p+1}$

r2 : $1^n + 1^m = 1^p$ |--- $1^n + 1^{m+1} = 1^{p+1}$

F1

F2

h

Fp : g

F'1

F'q : n

Ensemble des théorèmes
$\{1^n + 1^m = 1^{n+m}\}$

SYSTÈMES FORMELS

Définition

On appelle déduction à partir de $h_1,\ldots\ldots, h_n$ toute suite finie de formules

$f_1, \ldots, f_p$ telle que pour tout $i \in \{1, \ldots, p\}$

- f_i est un axiome

Ou bien

- f_i est l'une des formules $h_1, h_2, \ldots, h_n$

Ou bien

- f_i est obtenue par application d'une règle r_k à partir de formules $f_{i0}, \ldots, f_{il}$ placées avant f_i

On note : $h_1, h_2, \ldots, h_n \vdash_S f_p$

On appelle Théorème t «une déduction à partir de Ø» et on note $\vdash_S t$

{axiomes} ⊂ {théorèmes} ⊂ {mots} ⊂ {chaines de l'alphabet}

SYSTÈMES FORMELS

Terminologie

Syntaxe : Spécifie la manière de construire des expressions du langage

Dérivation : application d'une règle

Théorème : chaîne obtenue à partir des axiomes par une suite de dérivations

Preuve d'un théorème : séquence de dérivations pour l'obtenir

Système cohérent : système où il existe des formules bien formées qui ne sont pas des théorèmes

Système formel consistant : s'il n'existe pas de formule bien formée f telle que f et $\neg$ f soient des théorèmes

Système formel contradictoire : Système inconsistant

Système formel décidable : système où il existe une procédure de décision qui permet de décider, en un temps fini, si un mot quelconque du système est un théorème ou non-théorème

SYSTÈMES FORMELS

Exemple

Le (p q -) système :

- alphabet : { p, q, - }
- mots : suites finies de symboles de l'alphabet
- axiomes : les mots x p - q x - où x est formé uniquement de -
- une seule règle R : si x, y et z sont des mots formés uniquement de –

$$x\ p\ y\ q\ z \vdash_R x\ p\ y\ -\ q\ z\ -$$

Théorèmes ???

x p - q x - ⟶ $-^{a}\ \mathbf{p}\ -^{b}\ \mathbf{q}\ -^{a+b}$ **x p y q x y**

SYSTÈMES FORMELS

Exercice

Ensemble des déductions est récursif

On doit mettre en évidence un programme, qui pour toute suite finie d'éléments (f_1, …, f_n)de Σ^* indique au bout d'un temps fini si cette suite est une déduction ou pas.

Voici le programme :

(a)on commence par vérifier si chaque f_i est une fbf(on utilise le programme de F récursif)

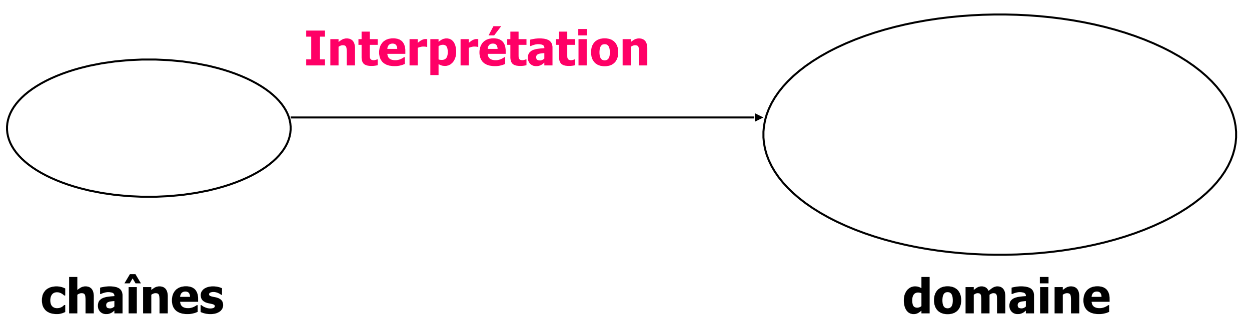
(b) on repère les axiomes (A est récursif)

(c) pour chaque formule, qui n'est pas un axiome, on vérifie qu'elle peut s'obtenir à partir des formules qui la précédent (en utilisant les programmes relatifs aux règles car elles sont décidables)

SYSTÈMES FORMELS

Sémantique:

- Donner un sens aux chaînes d'un système
- Choisir un domaine d'interprétation
- Associer à chaque chaîne une valeur d'un domaine



SYSTÈMES FORMELS

Interprétation

Une interprétation d'un système formel S = (Σ, F, A, R) est le choix d'un univers U de domaine D (D est un ensemble d'objets appartenant à U) et d'une application I : S → U qui associe :

à chaque symbole de (Σ, F, A, R)	un élément de D
au procédé de construction de mots F	un procédé de construction d'énoncés de U
à chaque axiome de A	un énoncé vrai de U
à chaque règle d'inférence de R	un mode de déduction dans U

Les théorèmes du système formel S deviennent des énoncés de U qu'on juge VRAIS ou FAUX dans l'interprétation (ou privés de sens)

LA LOGIQUE EN GÉNÉRAL

➢ les logiques sont des langages formels pour représenter l'information et permettre d'en tirer des conclusions

➢ Deux notions importantes:

Syntaxe: définit les expressions possibles du langage (i.e suite de mots et de symboles formant une phrase)

Sémantique: définit la signification d'une expression (sa valeur de vérité) dans un "monde" donné

➢ la logique offre un mécanisme de manipulation et de raisonnement sur l'information sémantique contenue dans une phrase en ne manipulant que les symboles qui la constituent

LA LOGIQUE EN GÉNÉRAL

Exemple

Dans le langage arithmétique:

- $x + 2 \geq y$ est une expression du langage,
- x2 + y > n'est pas une expression du langage,
- $x + 2 \geq y$ est vraie si et seulement si le nombre $x + 2$ n'est pas + petit que y,
- $x + 2 \geq y$ est vraie dans un monde où $x = 7$ et $y = 1$,
- $x + 2 \geq y$ est fausse dans un monde où $x= 0$ et $y= 6$

EXEMPLES DE LOGIQUES

Logique propositionnelle

Une suite de symboles séparés par des conjonctions (et), des disjonctions (ou) ou des négations (non)

Exemples :

« Socrate est un homme » $\Rightarrow$ homme(Socrate)

« Platon est un homme » $\Rightarrow$ homme(Platon)

« Platon et Socrate sont des hommes » $\Rightarrow$

homme(Socrate)^homme(Platon)

EXEMPLES DE LOGIQUES

Logique des prédicats du 1er ordre

Une suite de symboles, de variables et de relations avec des quantificateurs universels et existentiels

homme(Socrate) $\forall$ x homme(x) $\rightarrow$ mortel(x)

Autres logiques

Logique floue : accompagne les faits de valeurs de vraisemblance https://meet.google.com/xge-xxmm-dtn

LOGIQUE DES PROPOSITIONS

Le calcul des propositions ou logique des propositions a pour objectif l'étude des formes de raisonnement dont la validité est indépendante de la structure des propositions composantes et résulte uniquement de leurs propriétés d'être vraies ou fausses.

LOGIQUE DES PROPOSITIONS

Deux aspects complémentaires doivent être pris en compte et liés l'un à l'autre :

➢L'aspect syntaxique qui revient simplement à définir un système formel dans lequel les déductions qu'on peut faire conduisent à des théorèmes de calcul propositionnel

➢L'aspect sémantique qui est l'interprétation des formules et qui consiste en l'analyse des « formules toujours vraies » appelées tautologies

La liaison entre les deux aspects est la démonstration que les formules qui sont les tautologies (i.e. sémantiquement valables) sont les mêmes que les formules qui sont les théorèmes (i.e syntaxiquement valables)

LOGIQUE DES PROPOSITIONS

➢Une proposition est une expression (phrase) à propos du monde qui est soit vraie soit fausse.

➢Éléments de base:

- symboles de propositions: P, Q, … (phrases)

- phrases spéciales: Vrai, Faux

-opérateurs: ^ (et), v (ou), ⌉ (non), → (implique),

↔ (équivalent)

LOGIQUE DES PROPOSITIONS

➢Formation de phrases composées à l'aide des connecteurs:

- négation: $\rceil$ P
- conjonction: P ^ Q
- disjonction: P v Q
- implication: P $\rightarrow$ Q (si P alors Q)

-équivalence: P $\leftrightarrow$ Q (P et seulement si Q)

Exemples:

$\rceil (P \wedge Q) \vee (P \rightarrow (Q \leftrightarrow R))$ $\qquad$ $P \wedge Q \wedge (Q \rightarrow R)$

CALCUL PROPOSITIONNEL

DÉFINITION

On appelle calcul propositionnel P_0 le système formel défini par :

♦ $\sum = \{p_0, p_1, \ldots , p_n, \ldots\} \cup \{\rightarrow, \rceil, (,)\}$

les p_i sont appelées propositions atomiques, ou atomes

♦ $\mathcal{F}$ = le plus petit ensemble de formules tel que :

$\forall i \;\; p_i \in \mathcal{F}$

$\forall A \in \mathcal{F}, \forall B \in \mathcal{F} : \rceil A \in \mathcal{F}, (A \rightarrow B) \in \mathcal{F}$

CALCUL PROPOSITIONNEL

DÉFINITION

- ♦ $\mathcal{A}$ = ensemble de toutes les formules de l'une des trois formes suivantes :

  SA_1 : $(A \rightarrow (B \rightarrow A))$

  SA_2 : $((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$

  SA_3 : $((\rceil A \rightarrow \rceil B) \rightarrow (B \rightarrow A))$

- ♦ $\mathcal{R}$ = {m.p.}

  m.p.(modus ponens) : A , $(A \rightarrow B)$ |-- B

CALCUL PROPOSITIONNEL

Propositions

Proposition 1 :

Pour tout A de F (A→A) est un théorème

Proposition2

Si A1, ……. , An-1 |---- (An→B) **alors** A1, …… , An |--- B

Proposition3 (théorème de déduction) :

Si A1,…, An-1, An |---- B **alors** A1, … , An-1 |--- (An→B)

C[que] : A1,…, An-1, An |---- B **ssi** A1, … , An-1 |--- (An→B)

$A, (A \rightarrow B) \mid\!-\!-\ B$

$SA_1 : (A \rightarrow (B \rightarrow A))$

$SA_2 : ((A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C)))$

$SA_3 : ((\rceil A \rightarrow \rceil B) \rightarrow (B \rightarrow A))$

$\mid\!-\!-\ (A \rightarrow A)$?

F1 : $A \rightarrow (A \rightarrow A)$ (SA1)
F2 : $(A \rightarrow ((A \rightarrow A) \rightarrow A))$ (SA1)
F3 : $(A \rightarrow ((A \rightarrow A) \rightarrow A)) \rightarrow ((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$ (SA2)
F4 : $((A \rightarrow (A \rightarrow A)) \rightarrow (A \rightarrow A))$ m.p. F2 et F3
F5 : $(A \rightarrow A)$ m.p. F1 et F4

On a : A1, ……………., An-1 |— (An→B)

A-t-on A1, ……… , An-1, An |— B ?

F1 :

SAi et m.p et les hypothèses

Fp : (An→B)
Fp+1 : An hypothèse
Fp+2 : B par m.p. sur Fp+1 et Fp

Si A1,…, An-1, An |---- B **alors** A1, … , An-1 |--- (An→B)

A1, ……… , An-1, An|— B

F1 :

SAi , m.p et les hypothèses (Ai)i=1…n

Fk : B

F'1:

SAi , m.p et les hypothèses (Ai)i=1…n-1

F'q : (An→B)

Si A1,…, An-1, An |---- B **alors** A1, … , An-1 |--- (An→B)

F'1:

SAi , m.p et les hypothèses (Ai)i=1…n-1

F'q :(C→B)

F'q :C

F'k0 : B

CALCUL PROPOSITIONNEL

Démonstration du théorème de déduction

Par récurrence sur la longueur K de la déduction

Pour K=1

1er cas : B axiome

F1 : B

F2 : (B->(An->B)) (SA1)

F3 : (An->B) (m.p. F1 et F2)

2ème cas : B est l'une des hypothèses A1, ….. , An-1

F1 : B

F2 : (B->(An->B)) (SA1)

F3 : (An->B) (m .p. F1 et F2)

3ème cas : B est An

|-- (An->B) d'après la proposition 1 et donc A1, …… , An-1 |-- (An->B)

CALCUL PROPOSITIONNEL

Démonstration du théorème de déduction

On suppose la propriété vraie pour $k<k_0$

Il ya 4 cas possibles :

$SA_2 : ((A \to (B \to C)) \to ((A \to B) \to (A \to C)))$

(a)B axiome

(b) B = A1ou ….ou An-1

(c) B = An

(d) B est obtenu par le m.p. à partir de (C->B) et C

Les cas (a), (b) et (c) pareil que k=1

(C->B) et C sont des formules de la déduction B à partir des hypothèses $(A_i)_{1 \le i \le n}$

Donc :

A1, ….. , An|-- (C->B) (en moins de k_0 étapes)

A1, ….. , An|-- C (en moins de k_0 étapes)

L'hypothèse de récurrence donne que :

A1, ….. , An-1|— (An->(C->B))

A1, ….. , An-1|— (An->C)

Grâce à (SA2)

A1, ….. , An|-- ((An->(C->B))->((An->C)->(An->B))

((An->C)->(An->B)) par m.p

(An->B) par m.p

Exercice

$\vdash (A\to(B\to C))\to(B\to(A\to C))$?

$\vdash (A\to(B\to C))\to(B\to(A\to C))$ ssi
$(A\to(B\to C))\vdash (B\to(A\to C))$ ssi
$(A\to(B\to C)), B\vdash (A\to C)$ ssi
$(A\to(B\to C)), B, A\vdash C$

F1 : $A\to(B\to C)$ hypothèse
F2 : A hypothèse
F3 : $(B\to C)$ m.p. sur F1 et F2
F4 : B hypothèse
F5 : C m.p. F3 et F4

$\vdash (B\to((B\to C)\to C))$?

$\vdash (B\to((B\to C)\to C))$ ssi
$B\vdash (B\to C)\to C$ ssi
$B,(B\to C)\vdash C$

F1: B hypothèse
F2: $(B\to C)$ hypothèse
F3: C m.p. sur F1 et F2