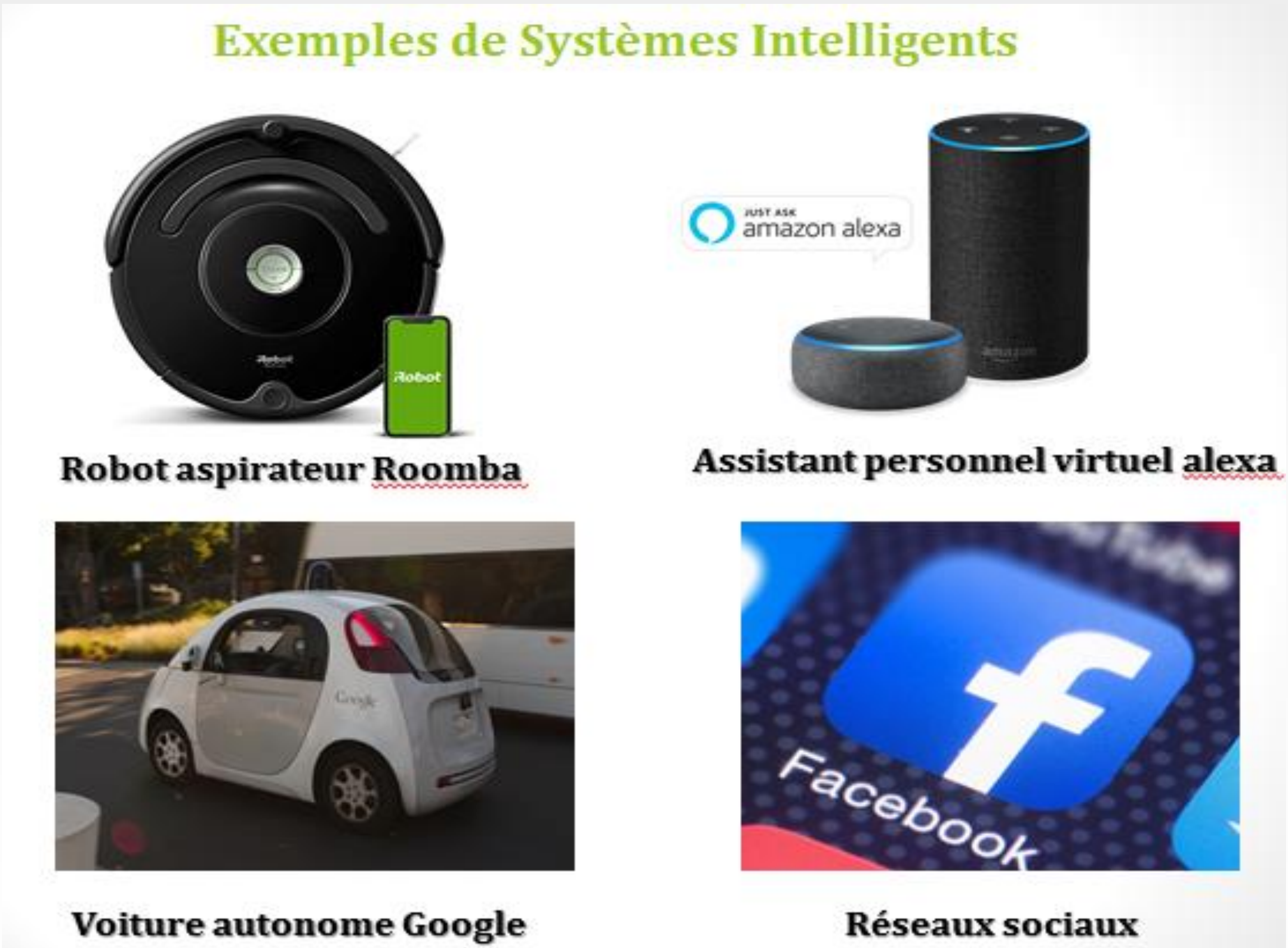
Systèmes Intelligents ?



Système Intelligent: Synthèse

Pour être qualifié d'intelligent, une application/machine doit avoir des facultés comme:

- Traitement du langage naturel (pour lui permettre de communiquer avec succès)
- Apprentissage (pour s'adapter à de nouvelles circonstances et pour détecter et extrapoler les modèles)
- Raisonnement approximatif en présence de l'incertitude

Chapitre 3

- Représentation des connaissances (pour stocker ce qu'il sait)
- Raisonnement automatisé (pour utiliser les informations stockées pour répondre aux questions et tirer de nouvelles conclusions)

Chapitre 1 & 2

Aussi on peut penser aux domaines de la:

- Vision par ordinateur (pour percevoir les objets)
- Robotique (pour manipuler des objets et se déplacer)

Université de Jijel
Faculté des Sciences Exactes et Informatique
Département d'Informatique

Systèmes Intelligents

CHAPITRE I

Logiques de Description (LD)

Octobre 2021

1. Introduction

Représentation de la connaissance

- Un système intelligent est un système qui peut imiter, automatiser certains comportements intelligents de l'être humain. [1]

- Un comportement intelligent dépend généralement de la disponibilité de la connaissance appropriée et de la capacité de tirer des conclusions à partir de cette connaissance et des faits observés.

  - Pour cette raison, la représentation des connaissances et le raisonnement sont un sous-domaine clé de l'IA.

Représentation de la connaissance

- L'objectif général de la représentation des connaissances est de **développer des formalismes** pour fournir des descriptions de haut niveau du monde qui peuvent être utilisé efficacement pour créer des **applications intelligentes**.

Formalismes de représentation de connaissance ?

Formalismes de représentation de connaissance : exemples

Formalismes Logiques

- Logique propositionnelle
- Logique des prédicats
- Logiques modales (modalités: nécessité/ possibilité, savoir/croyance, passé/future)
- Logique floue

Autres Formalismes

- Réseaux sémantiques (Représentation graphique)
- Règles de production (Si – Alors)
- Représentation objet (Frames ou schémas)
- Modèles graphiques probabilistes (Réseaux bayésiens)
- ….

Formalismes logiques : Logique des propositions

Imaginons un domaine de connaissance composé de :

- trois personnes (Mark, Tim, Bill),
- deux propriétés (Riche, Content)
- une relation binaire (sont_Amis).

Combien de variables propositionnelles a-t-on besoin pour représenter nos connaissances sur ce domaine?

« Une telle personne vérifie une telle propriété »
« Une telle personne est en relation avec une autre personne »

Formalismes logiques : Logique des propositions

« Une telle personne vérifie une telle propriété »

Mark est riche p_{11}
Mark est content p_{12}
Tim est riche p_{21}
Tim est content p_{22}
Bill est riche p_{31}
Bill est content p_{32}

« Une telle personne est en relation avec une autre personne »

Mark et Tim sont des amis r_{12}
Mark et Bill sont des amis r_{13}
Tim et Bill sont des amis r_{23}

9= 3 x 2 + (2+1)

Nbr des objets * Nbr des propriétés + ((Nbr des objets -1)+….+1) * Nbr de relations

Formalismes logiques : Logique des propositions

- Au moins une personne est contente :

$$p_{12} \vee p_{22} \vee p_{32}$$

- Il y a au moins deux personnes contentes et qui sont des amies

$$(p_{12} \wedge p_{22} \wedge r_{12}) \vee (p_{12} \wedge p_{32} \wedge r_{13}) \vee (p_{22} \wedge p_{32} \wedge r_{23})$$

- Refaire les mêmes questions pour: 10 objets, 15 propriétés et 15 relations

$$ça\ reste\ faisable, mais\ ça\ devient\ très\ compliqué!!$$

- On ne sait pas combien il y a de personnes. Représenter l'énoncé suivant: « **Toute personne est soit riche ou contente** »

$$Impossible\ \ldots.$$

Formalismes logiques : Logique des propositions

- **Expressivité très limitée** avec seulement des propositions atomiques et des connecteurs booléens. Comment :
  - Représenter **les relations** entres objets dans le mode
  - Exprimer **la quantification**
- Logique **décidable**: c'est-à-dire qu'il existe une **procédure effective** qui peut montrer (*en un nombre fini d'opérations = en un temps fini*) qu'une formule est valide ou contradictoire.

Formalismes logiques : Logique des prédicats

- **Très expressive** (Prédicats, quantificateurs existentielle et universel, variables, constantes) on se trouve des fois devant des expressions logiques volumineuses.
- **Indécidable** ou plus précisément **semi-décidable** c'est-à-dire ::
  - **Si** une formule est **valide** alors la procédure s'arrête et retourne **`oui'**
  - **Sinon** ou bien la procédure s'arrête et retourne `non', ou bien elle ne s'arrête pas.

Formalismes logiques : Logiques de Description

- Une **famille** de langages de représentation des connaissances :
  - Les notions importantes du domaine de connaissance sont représentées par des **descriptions de concepts**
  - Dotées d'une **sémantique formelle basé sur la logique** (différemment de leurs prédécesseurs, tels que les réseaux sémantiques et les frames).

LD : Syntaxe bien définie et Sémantique formelle non ambiguë

Formalismes logiques : Logiques de Description

- L'enjeu était de trouver un **compromis** entre expressivité et décidabilité.
- Les tâches de raisonnement en LD sont:
  - ✓ **Décidables** : elles peuvent toujours être correctement exécutées en un **temps fini**.
  - ✓ **Facile** (tractable): elles peuvent toujours être correctement complétés dans un **temps polynomial** par rapport à la taille de la BC.

La **majorité** des LDs sont des **fragments décidables** de la logique des prédicats.

Domaines d'application des LDs

- Bases de données distribuées
- Traitement automatique du langage naturel
- …
- Récemment Web **sémantique** : sont adoptées comme base pour les **langages d'ontologie** tels que OWL et ses prédécesseurs OIL et DAML+OIL. les bases de connaissances LD sont maintenant souvent appelées **ontologies**.

2. Logiques de Description

Les logiques de description séparent les connaissances d'un domaine en deux composantes:

T (Tbox – Terminological part**)**

- Représente la **t**erminologie d'un domaine
- Connaissance sur la **structure** d'un domaine

A (Abox – Assertional part**)**

- Contient les **a**ssertions sur des individus nommés du domaine considéré
- Connaissance sur une situation concrète

La combinaison TBox et ABox étant appelée base de connaissances (BC) et noté souvent **Σ** .

Un premier Exemple

En langage naturelle

Un enseignant est une personne qui enseigne un cours.

Ali est une personne , IA est un cours,   Ali enseigne IA

En logique des prédicats

$$\forall x \left(Enseignant(x) \Leftrightarrow Personne(x) \wedge \exists y \left(enseigne(x, y) \wedge Cours\ (y) \right) \right)$$

$Personne(Ali)$

$Cours\ (IA)$

$enseigne\ (Ali, IA)$

En LD:

$$\left.\text{Enseignant} \equiv \text{Personne} \ \cap \exists \text{enseigne.cours}\right] \quad \text{TBOX}$$

Ali : Personne,

IA : Cours

(Ali, IA) : $enseigne$

ABOX

Deux types de descriptions:

- Descriptions élémentaires: concepts atomiques et les rôles atomiques
- Descriptions complexes: peuvent être construites en utilisant Descriptions élémentaires & constructeurs .

Enseignant ≡ Personne ∩ ∃ enseigne. cours

Personne, cours: **concepts atomiques**

enseigne **: rôle atomique**

Enseignant : Description **complexe**

Sémantique LD

Elle est donnée au moyen d'un domaine d'interprétation Δ^I qui est un ensemble non vide d'individus et d'une fonction d'interprétation **I** qui fait correspondre à :

- un concept un sous-ensemble de Δ^I
- un rôle un sous-ensemble de Δ^I X Δ^I

3. Syntaxe et sémantique des LDs

Logique AL (Attributive Language)

Constructeur	Syntaxe	Sémantique
Concept atomique	A	$A^I \subseteq \Delta^I$
Rôle atomique	R	$R^I \subseteq \Delta^I X \Delta^I$
Concept le plus générale	T	Δ^I
Concept le plus spécifique	⊥	$\emptyset$
Négation **atomique**	$\neg A$	$\Delta^I \backslash A^I$
Conjonction	$C \cap D$	$C^I \cap D^I$
Quantification universelle	$\forall R.C$	$\{a \in \Delta^I \mid \forall b(a,b) \in R^I \rightarrow b \in C^I\}$
Quantification existentielle non qualifiée	$\exists R.T$	$\{a \in \Delta^I \mid \exists b(a,b) \in R^I\}$

Logique AL (Attributive Language)

Constructeur	Syntaxe	Sémantique
Concept atomique	A	$A^I \subseteq \Delta^I$
Rôle atomique	R	$R^I \subseteq \Delta^I X \Delta^I$
Concept le plus générale	T	Δ^I
Concept le plus spécifique	⊥	$\emptyset$
Négation **atomique**	$\neg A$	$\Delta^I \backslash A^I$
Conjonction	$C \cap D$	$C^I \cap D^I$
Quantification **universelle**	$\forall R.C$	$\{a \in \Delta^I \mid \forall b(a,b) \in R^I \rightarrow b \in C^I\}$
Quantification existentielle **non qualifiée**	$\exists R.T$	$\{a \in \Delta^I \mid \exists b(a,b) \in R^I\}$

Autres constructeurs

U	Disjonction ou union de deux concepts	$C \cup D$	$C^I \cup D^I$
$\mathcal{E}$	Quantification existentielle	$\exists R.C$	$\{a \in \Delta^I \mid \exists b (a,b) \in R^I \wedge b \in C^I\}$
$\aleph$	Restriction de nombre	$\leq nR$ $\geq nR$ $= nR$	$\{a \in \Delta^I \mid \lvert\{b \mid (a,b) \in R^I\}\rvert \geq n\}$ $\{a \in \Delta^I \mid \lvert\{b \mid (a,b) \in R^I\}\rvert \leq n\}$ $\{a \in \Delta^I \mid \lvert\{b \mid (a,b) \in R^I\}\rvert = n\}$
F	Rôle fonctionnel	$\leq 1\ R$	$\{a \in \Delta^I \mid \lvert\{b \mid (a,b) \in R^I\}\rvert \leq 1\}$
C	Négation de concepts arbitraire	$\neg C$	$\Delta^I \backslash C^I$
Q	Restriction de nombre qualifiée	$\leq nR.C$ $\geq nR.C$ $= nR.C$	$\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I \wedge b \in C^I\}\rvert \leq n\}$ $\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I \wedge b \in C^I\}\rvert \geq n\}$ $\{a \in \Delta^I \mid \lvert\{b \in \Delta^I \mid (a,b) \in R^I \wedge b \in C^I\}\rvert = n\}$
S	Clôture transitive de rôle	R^+	$\bigcup_{i \geq 1} (R^I)^i$
H	Rôle hiérarchique	$R \subseteq S$	$R^i \subseteq S^i$
I	Rôle inverse	R^-	$\{(b,a) \in \Delta^I \times \Delta^I \mid (a,b) \in R^I\}$
O	Description des concepts par énumération d'individus nommés	$\{a_1\} \cup \ldots \{a_n\}$	$\{a_1{}^I\} \cup \ldots \cup \{a_n{}^I\}$

Logiques de Description: exemples

- AL + la négation de concepts arbitraires = **ALC**
- ALC + H + S = la logique **SH**.
- SH + I + F = la logique **SHIF**
- SH + O+ I + N = logique **SHOIN**
- **….**

❑ TBOX est un ensemble fini d’axiomes terminologiques :

Axiome	Lecture
$C \equiv D$	Le Concept C est par définition égal au concept D (on utilise la même syntaxe pour définir l’équivalence entre rôles)
$C \subseteq D$	Le Concept C est par définition subsumé par le concept D (on utilise la même syntaxe pour exprimer la subsomption entre rôles)

NB. $C \equiv D$ signifie que $C \subseteq D$ et $D \subseteq C$

❑ ABOX est un ensemble fini d’**assertions**:

Assertion	Lecture
a: C	a est un individu du concept C
(a, b): R	Les individus a et b sont liés par R

Propriétés

- **Satisfiabilité (consistance):** Un concept C est satisfaisable, pour une interprétation I, SSI $C^I \neq \phi$.
- **Subsomption:** Un concept C est subsumé par un concept D, pour une interprétation I, SSI $C^I \subseteq D^I$
- **Equivalence :** Un concept C est équivalent à un concept D, pour une interprétation I, SSI $C^I = D^I$
- **Incompatibilité ou disjonction:** Deux concepts C et D sont incompatibles, pour une interprétation I, SSI $C^I \cap D^I = \phi$

Travaux Dirigés

Exercice 1: Du langage naturel vers la logique de description

Exercice 2: De la logique des prédicats vers la logique de description

Exercice 3: Sémantique

Exercice 4: Application sur les propriétés

Problèmes d'inférences

Etant donnée une BC en LD, les problèmes d'inférence que nous souhaitons résoudre sont les suivants (on parle de concepts ou de classes indifféremment):

1. Satisfiabilité de la BC dans l'ensemble (possède-t-elle un modèle ?)
2. Satisfiabilité de classes.
3. Disjonction de classes.
4. Inclusion (ou subsomption) de classes.
5. Equivalence de classes.
6. Appartenance d'un individu à une classe.
7. Recherche de tous les individus connus d'une classe.

Problèmes d'inférences

- Les problèmes de 2 à 7 se réduisent au problème 1 (Satisfiabilité de la BC)
- Le problème 1 peut être résolu en utilisant l'algorithme des **Tableaux Sémantiques**.
- Cet algorithme est basé sur l'idée de **preuve par réfutation** :

  « afin de prouver qu'une **formule est satisfiable** on montre que sa négation est une **contradiction»**

Réduction au non satisfiabilité

Non satisfiabilité d'une classe
$(K \vDash \boldsymbol{C} \subseteq \bot)$ SSI $K \cup \{\boldsymbol{C}\ (\boldsymbol{a})\}$ est **non satisfiable** tel que a est un nouvel individu (ne figure pas déjà dans K).

Mais, on cherche à montrer la Satisfiabilité de classes ??

Réduction au non satisfiabilité

Non satisfiabilité d'une classe
$(K \vDash \boldsymbol{C} \subseteq \bot)$ SSI $K \cup \{\boldsymbol{C}(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un nouvel individu (ne figure pas déjà dans K).

Disjonction de classes
$(K \vDash \boldsymbol{C} \cap \boldsymbol{D} \subseteq \bot)$ SSI $K \cup \{(\boldsymbol{C} \cap \boldsymbol{D})(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un

Réduction au non satisfiabilité

Non satisfiabilité d'une classe
$(K \vDash \boldsymbol{C} \subseteq \perp)$ SSI $K \cup \{\boldsymbol{C}(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un nouvel individu (ne figure pas déjà dans K).

Disjonction de classes
$(K \vDash \boldsymbol{C} \cap \boldsymbol{D} \subseteq \perp)$ SSI $K \cup \{(\boldsymbol{C} \cap \boldsymbol{D})(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un

Subsomption
$(K \vDash \boldsymbol{C} \subseteq \boldsymbol{D})$ SSI $K \cup \{(\boldsymbol{C} \cap \neg\boldsymbol{D})(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un nouvel individu .

Equivalence de classes
$(K \vDash \boldsymbol{C} \equiv \boldsymbol{D})$ SSI $K \vDash \boldsymbol{C} \subseteq \boldsymbol{D}$ et $K \vDash \boldsymbol{D} \subseteq \boldsymbol{C}$

Réduction au non satisfiabilité

Non satisfiabilité d'une classe
$(K \vDash \boldsymbol{C} \subseteq \perp)$ SSI $K \cup \{\boldsymbol{C}(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un nouvel individu (ne figure pas déjà dans K).

Disjonction de classes
$(K \vDash \boldsymbol{C} \cap \boldsymbol{D} \subseteq \perp)$ SSI $K \cup \{(\boldsymbol{C} \cap \boldsymbol{D})(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un

Subsomption
$(K \vDash \boldsymbol{C} \subseteq \boldsymbol{D})$ SSI $K \cup \{(\boldsymbol{C} \cap \neg \boldsymbol{D})(\boldsymbol{a})\}$ est **non satisfiable** tel que a est un nouvel individu .

Equivalence de classes
$(K \vDash \boldsymbol{C} \equiv \boldsymbol{D})$ SSI $K \vDash \boldsymbol{C} \subseteq \boldsymbol{D}$ et $K \vDash \boldsymbol{D} \subseteq \boldsymbol{C}$

Appartenance d'un individu à une classe
$(K \vDash \mathbf{C}(\boldsymbol{a}))$ SSI $K \cup \{\neg \boldsymbol{C}(\boldsymbol{a})\}$ est **non satisfiable**.

Recherche de tous les individus connus d'une classe
Il faut vérifier pour **tous les individus** a, si $\boldsymbol{K} \vDash \boldsymbol{C}(\boldsymbol{a})$.

- Avant d'appliquer l'algorithme des tableaux sémantiques, il faut réécrire la BC en utilisant seulement les symboles **de l'union et de l'intersection**.
- Ensuite, il faut transformer la BC à son équivalent en NNF (*Negation Normal Form*) comme suit :

$$
\begin{aligned}
&NNF(C) = C \text{ si } C \text{ est un concept atomique}\\
&NNF(\neg C) = \neg C \text{ si } C \text{ est un concept atomique}\\
&NNF(\neg\neg C) = NNF(C)\\
&NNF(C \cup D) = NNF(C) \cup NNF(D)\\
&NNF(C \cap D) = NNF(C) \cap NNF(D)\\
&NNF(\neg(C \cup D)) = NNF(\neg C) \cap NNF(\neg D)\\
&NNF(\neg(C \cap D)) = NNF(\neg C) \cup NNF(\neg D)\\
&NNF(\forall R.C) = \forall R.NNF(C)\\
&NNF(\exists R.C) = \exists R.NNF(C)\\
&NNF(\neg\forall R.C) = \exists R.NNF(\neg C)\\
&NNF(\neg\exists R.C) = \forall R.NNF(\neg C)
\end{aligned}
$$

Algorithme des Tableaux Sémantiques pour la logique ALC

Soit $K = (T, A)$ et A est consistante.

1 S ← {A}.

2 S'il y a une règle à appliquer Aller vers l'étape 3.
Sinon K est consistante ; Exit.

3 Appliquer la règle sur S.

4 Supprimer tous les **A'** contenant C(a) et $\neg C(a)$.

5 Si $S = \emptyset$ alors K est inconsistante ; Exit.
Sinon aller à l'étape 2.

Quelques règles d'extension

Soit $A' \in S$:

Règle d'instanciation

Condition :

$C \in T$

Action :

Ajouter C(a) à A′ tel que : a est un individu connu (*mentionné dans A′*)

$\cap$ ***–Règle***

Condition :

$(C \cap D)$ (a) $\in \mathbf{A'}$ et que $C(a)$ et $D(a)$ n'appartiennent pas ***à la fois*** à $\mathbf{A'}$

Action :

Supprimer A′ et ajouter $A' \cup \{C(a), D(a)\}$ à S

$\cup$ **– Règle**

Condition :

$(C \cup D)$ (a) $\in \mathbf{A'}$ et que **ni** $C(a) \in \mathbf{A'}$ **ni** $D(a) \in \mathbf{A'}$

Action :

Supprimer A′ de S ensuite ajouter $A' \cup \{C(a)\}$ et $A' \cup \{D(a)\}$ à S.

Bibliographie