

ONTOLOGIES ET LOGIQUES DE DESCRIPTION

□ Objectif du cours:

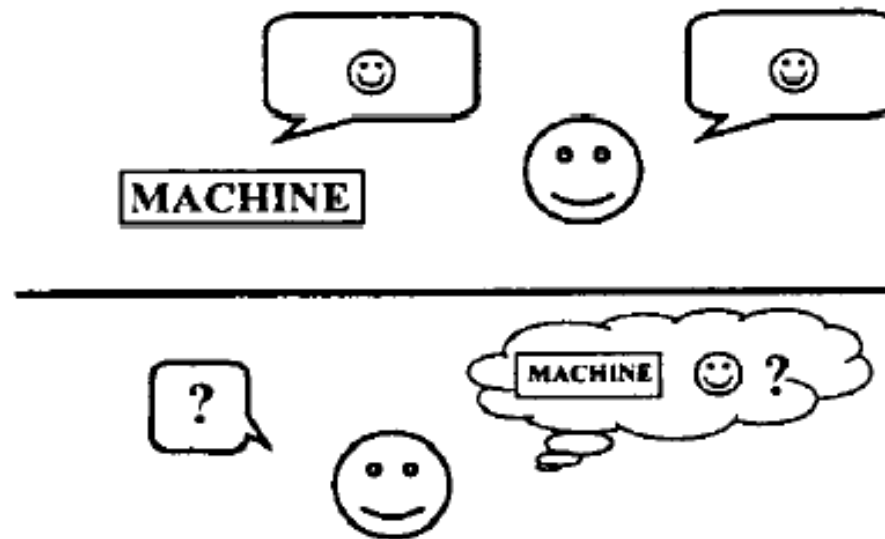
- Comprendre les concepts et formalismes qui sous-tendent le web sémantique (Logiques de Description, Ontologies)
- Connaitre les outils et langage de mise en œuvre
 - DAML+OIL
 - OWL
 - Protégé

Motivations

4

Can machines think?

(A. Turing. "Computing Machinery and intelligence" , 1950)



Test de Turing pour l'intelligence

Motivations

5

Objectif : Simuler le raisonnement humain

- Le raisonnement humain est basé sur :
 - ▣ La représentation que le cerveau se fait des connaissances acquises (et stockées)
 - ▣ Les mécanismes logiques de déductions de nouvelles connaissances à partir de celles déjà représentées

- Pour simuler le raisonnement humain en machine, Il faut donc se donner la possibilité de :
 - ▣ représenter la connaissance en machine
 - ▣ raisonner sur les connaissances représentées (Inférence)

Motivations

6

- Hypothèse forte en IA:
 - Tout comportement intelligent requiert une quantité importante de connaissances.
 - En conséquence tout système logiciel sensé montrer un certain niveau d'intelligence doit offrir les moyens de **représenter** correctement les connaissances et les mécanismes pour **raisonner** avec ces connaissances.

Connaissance?

7

- **L'inférence** : l'élément de base de toute description de la cognition
 - Pour être efficace, l'inférence doit être guidée par la **connaissance** => capacité à mobiliser des informations pour agir
 - Le passage de INFORMATION à CONNAISSANCE est lié à l'expérience de l'action => pas de frontière parfaitement définie
 - Essai de définition : Connaissance = Information (donnée) qui influence un processus.

Représentation ? [2]

8

- Représenter \Leftrightarrow **Approximer** dans le contexte d'une tâche (activité?) particulière
- Représenter \Leftrightarrow Structure de **symboles** pour « décrire » une approximation du « monde » (un **modèle** du monde) dans le contexte d'une tâche particulière.
- Interpréter une structure (une représentation) \Leftrightarrow **Composition** de l'interprétation des différents symboles la constituant

Langage de représentation

9

Quel langage pour représenter les connaissances ?

- Langage naturel (français, anglais... etc.) trop ambiguë
- Langages formels OK (\Rightarrow Formalismes de RC):
 - ▣ logique du 1^{er} ordre (LPO), réseaux sémantiques, frames, logiques de description (LD)...etc.
 - ▣ **Qualités** : Adéquation représentative et inférentielle, efficacité inférentielle, efficacité acquisitionnelle, dissonance réduite.

RC dans le contexte du Web Sémantique

11

- Le Web est un media de partage de documents entre des personnes
- Comment ajouter à ces documents des données destinées aux ordinateurs?
 - ▣ Intégrer des documents avec des marqueurs sémantiques destinés à la machine
 - ▣ Trouver le sens des données sémantiques en suivant des hyperliens menant à la définition des termes clé et aux règles pour raisonner logiquement avec ces données
 - ▣ Soutenir le développement de services web automatiques et d'agents logiciels hautement fonctionnels.
- **Problème: Comment représenter les connaissances dans le contexte du Web**

Représentation des Connaissances dans le contexte du Web Sémantique

□ Technologies basiques du Web

- Uniform Resource Identifier (URI) :
 - Identifier les items sur le Web
- Extensible Markup Language (XML)
 - Permettre à chacun de définir son propre format de document
 - Peut intégrer des markups pour permettre la compréhension du contenu du document
- Resource Description Framework (RDF)
 - Faire des instructions exécutables par la machine
 - Triplet d'URIs (sujet, prédicat, objet)

Représentation des Connaissances dans le contexte du Web Sémantique

- **Comment faire dériver des nouvelles connaissances**
 - Nécessité d'avoir une sémantique formelle
 - Il existe plusieurs logiques
 - Les recherches doivent être décidable
 - La décidabilité s'oppose souvent à l'expressivité
 - Différentes applications nécessitent souvent différentes expressivités
 - D'une simple structure hiérarchique à des expressions logiques avec la négation et la quantification
 - Un couplage étroit entre la théorie et la pratique

Représentation des Connaissances dans le contexte du Web Sémantique

□ Comment faire dériver des nouvelles connaissances ?

□ Logique des Prédicats du Premier Ordre (LPPO)

■ Plus :

- très expressive et puissant

■ Moins:

- Pas très intuitif, modélisation difficile
- Complexité du calcul (indécidable dans le pire des cas)
- EXPTIME (exponentielle) souvent

Représentation des Connaissances dans le contexte du Web Sémantique

□ Comment faire dériver des nouvelles connaissances ?

□ Logique de Horn

■ Plus :

- Complexité gérable

■ Moins:

- Moins expressive
- Modélisation peu intuitive

Représentation des Connaissances dans le contexte du Web Sémantique

- **Comment faire dériver des nouvelles connaissances ?**
 - **Frames et Réseaux Sémantiques**
 - Plus :
 - Plus intuitif
 - Centré humain
 - Moins:
 - Manque de sémantique formelle nécessaire pour le raisonnement

Logiques de Description

18

□ Caractéristiques

- Les Logiques de Description (LD) Proviennent d'une combinaison des frames sémantiques et de la LPPO
- Les LD peuvent être définies à différents niveau d'expressivité en utilisant différentes caractéristique de la LPPO
 - Différentes classes d'expression correspondent à différents fragments de la LPPO
 - Plus d'expressivité → plus de complexité
 - Les sous-ensemble des DL sont appelés des *Langages de Description*
 - Les langages de description sont Décidables
- Ont donné lieu par implementation à plusieurs langages: **RDF+RDF/S** , **OWL**, **DAML+OIL**

Logiques de Description

19

□ Remarque :

- Définir les différentes catégories d'individus dans un domaine, ainsi que les relations existantes entre elles, permet de capturer les connaissances dans ce domaine.

- Origine des LD : les réseaux sémantiques (RS)

- Les RS apportent une vue graphique des bases de connaissances en organisant leurs contenus sous forme de réseau.
 - **Nœuds du réseau** : les classes d'individus, parfois les individus eux-mêmes.
 - **Liens** : les relations entre ces classes ou individus.

Logiques de Description

20

□ Exemple de RS

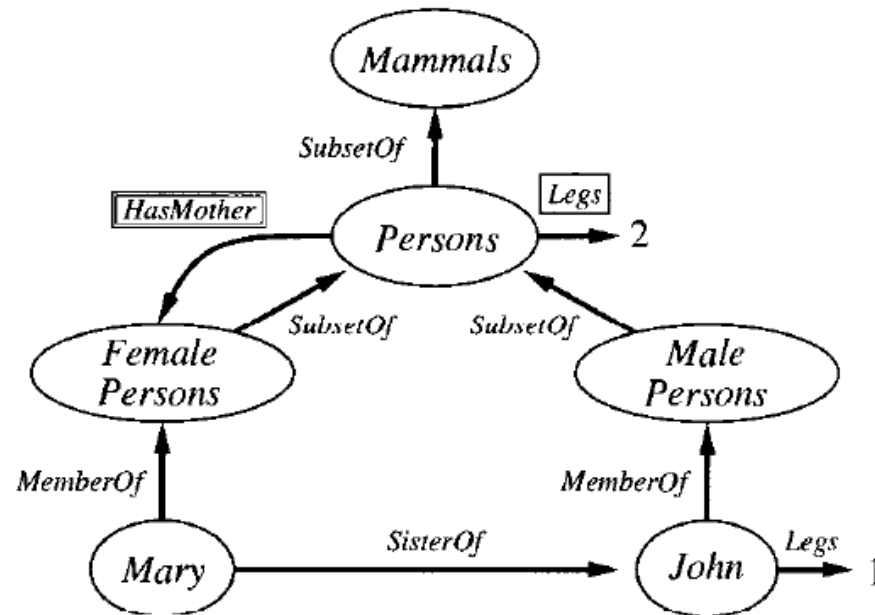


Fig 1 : Réseau sémantique avec 4 objets (Mary, John, 1, 2) et 4 classes d'individus. Les relations sont représentées par les liens étiquetés. (**Source**. "artificial Intelligence, A modern approach")

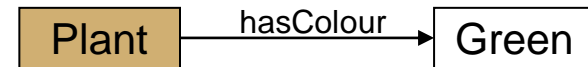
- Ils fournissent des algorithmes efficaces pour déduire les propriétés d'un objet sur la base de son appartenance à une classe.

Logiques de Description

21

Les réseaux sémantiques ainsi que d'autres représentations à objets sont ambiguës du point de vue logique.

Ex: Quelle est la signification exacte de :



- Chaque plante est uniquement de couleur verte
 - Les plantes sont aussi de couleur verte
 - Les plantes sont de couleur verte mais il peut y avoir des exceptions
-
- **Les LD** sont une réponse au besoin de **formaliser les représentations graphiques** qu'offrent les réseaux sémantiques tout en conservant l'accent qui est mis sur la hiérarchisation des classes d'individus.

Pourquoi pas la LPO ?

- Trop expressif pour obtenir des problèmes d'inférence décidables et efficaces

Logiques de Description

22

□ Base de Connaissances en LD

- Dans une BDC construite avec une LD, on distingue trois types d'objets formels:

- Les **concepts** :

- description de structures, représenté sous forme de prédicats unaires.
- Chaque concept représente un ensemble d'individus
- Les concepts sont intégrés dans une structure hiérarchique

- Les **rôles** :

- description des relations entre concepts
- Représenté par des prédicats binaires.

- Les **individus** :

- objets bien identifiés (nommés) du domaine.
- Membres des concepts.
- Peuvent être énumérés récursivement

Logiques de Description

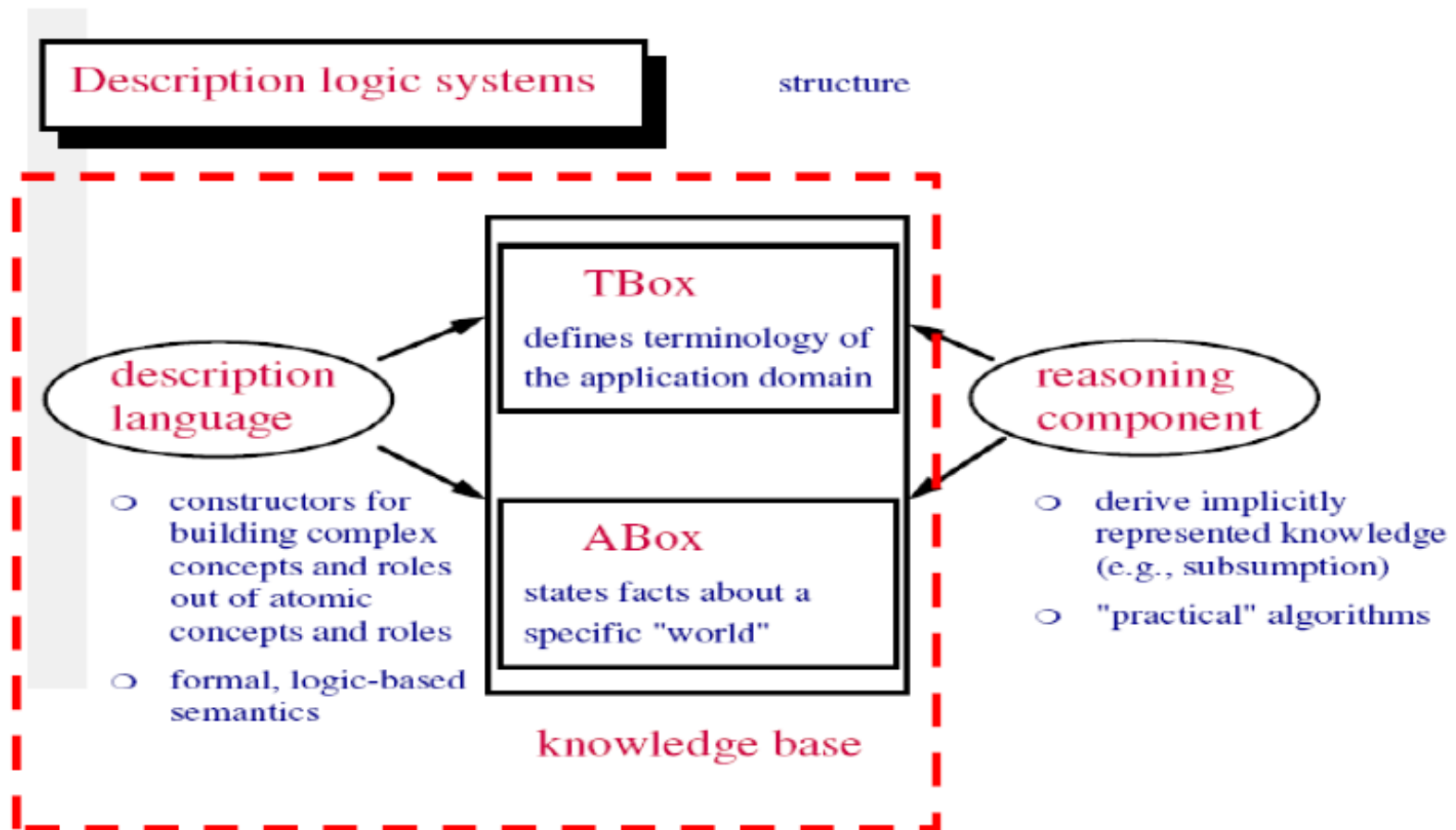
23

- **Base de Connaissances en LD**
- Une telle BDC contient deux parties :
 - ▣ La **TBox** (Term Box) qui introduit la terminologie du domaine : description des classes d'objets (concept), du vocabulaire du domaine.
 - ▣ La **Abox** (Assertional Box) qui contient les informations (assertions) sur les individus nommés du domaine, en terme du vocabulaire décrit dans la Tbox. Les assertions sont de la forme $C(a)$, $R(a,b)$.

Logiques de Description

24

Les systèmes à base de LD



Les familles de Langages de Description

25

□ La Tbox

- Définit le vocabulaire de la BDC
 - Le Langage de Description utilisé contrôle la complexité
 - Fournit les fondements théorique pour le raisonnement
 - Définit les concepts et les rôles complexes

□ Plusieurs Langages de description

- Différents par leurs attributs
- Chaque attribut augmente l'expressivité et la complexité
- Les Langages de Description sont nommés et classifiés par leurs attributs
- Langage le plus basique : *AL*

Les familles de Langages de Description

26

- **Hypothèses communes aux langages de Description**
 - **Hypothèse du Monde Ouvert:**
 - Le domaine d'interprétation Δ^I est infini
 - L'absence de connaissance sur un fait ne signifie pas que le fait contraire est vrai

 - **Hypothèse de l'ambiguïté des noms**
 - Deux concepts portant des noms différents peuvent être équivalents
 - Des noms différents ne garantissent pas que les concepts sont différents

Les familles de Langages de Description

27

□ Syntaxe

□ Pas de Variable

□ Plus simple que la syntaxe de la LPPO

■ LD : $\geq 4 R$

■ LPPO: $\exists y_1, y_2, y_3, y_4 (R(x, y_1) \wedge R(x, y_2) \wedge R(x, y_3) \wedge R(x, y_4) \wedge y_1 \neq y_2$
 $\wedge y_1 \neq y_3 \wedge y_1 \neq y_4 \wedge y_2 \neq y_3 \wedge y_2 \neq y_4 \wedge y_3 \neq y_4)$

Langage de Description *AL* (Attribute Language)

28

□ Langage de Description *AL*

□ Langage minimal

□ **Constructeurs syntaxiques autorisés et leurs interprétations:**

■ **Concepts Atomiques** (dénotés par A et B)

$$\blacksquare A^I \subseteq \Delta^I, \quad B^I \subseteq \Delta^I$$

■ **Rôles Atomiques** (dénotés par R)

$$\blacksquare R^I \subseteq \Delta^I \times \Delta^I$$

■ **Concepts Complexes** (dénotés par C et D)

$$\blacksquare C^I \subseteq \Delta^I, \quad D^I \subseteq \Delta^I$$

Langage de Description *AL*

29

□ Langage de Description *AL*

□ Constructeurs syntaxiques autorisés et leurs interprétations (suite):

- **T : Top** ou **Concept Universel**; représente le domaine contenant tous les individus

- $T^I = \Delta^I$

- **\perp : Bottom Concept (concept vide)**; représente le domaine ne contenant aucun individu;

- $\perp^I = \emptyset$

- **$\neg A$: Négation atomique;**

- La négation dans *AL* s'applique seulement aux concepts atomiques;

- A cause de l'hypothèse du monde ouvert, l'ensemble résultant est infini

- $(\neg A)^I = \Delta^I \setminus A^I$

Langage de Description *AL*

30

□ Langage de Description *AL*

□ Constructeurs syntaxiques autorisés et leurs interprétations (suite):

■ $C \sqcap D$: Intersection

- Tous les individus qui sont à la fois C et D
- $(C \sqcap D)^I = C^I \cap D^I$

■ $\forall R \bullet C$ Restriction de Valeurs

- Définit l'ensemble des individus qui sont en relation R avec des individus de C (et seulement eux);
- $\forall \text{etudie} \bullet \text{EcoleIngenieur}$: tous les individus qui étudient seulement dans une école d'ingénieurs.
- $(\forall R \bullet C)^I = \{a \in \Delta^I \mid \forall b ((a, b) \in R^I \rightarrow b \in C^I)\}$

Langage de Description *AL*

31

□ Langage de Description *AL*

□ Constructeurs syntaxiques autorisés et leurs interprétations (suite):

■ $\exists R \bullet T$ Restriction existentielle limitée

- Définit l'ensemble des individus qui sont en relation R avec un individu au moins
- \exists étudie $\bullet T$: tous les individus qui étudient
- $(\exists R \bullet T)^I = \{a \in \Delta^I \mid \exists b ((a, b) \in R^I)\}$

Langage de Description *AL*

32

□ Langage de Description *AL*

□ Axiomes Terminologiques

- La Tbox contient des affirmations sur les concepts complexes sous la forme d'axiomes terminologiques
 - **$C \subseteq D$: Inclusion, subsumption**
 - C est inclus dans D (est un sous-concept de D)
 - Chaque individu de C est également individu de D
 - Fournit des informations supplémentaires sur les relations entre les concepts et les rôles
 - $C^I \subseteq D^I$

Langage de Description *AL*

33

□ Langage de Description *AL*

□ Axiomes Terminologiques

■ $C \equiv D$: Equivalence

- C est équivalent à D (les deux identifient les mêmes individus)
- Chaque individu de C est également individu de D et vice-versa
- L'équivalence peut être utilisée pour définir de nouveaux concepts

- $C' = D'$

Langage de Description *AL*

34

□ Langage de Description *AL*

□ Axiomes Terminologiques

- Les axiomes terminologiques peuvent être utilisés pour modéliser et vérifier les assertions entre classes:
 - Exemple : classes disjointes : $\mathbf{C} \sqcap \mathbf{D} \equiv \perp$
 - La vérification des assertions concernent:
 - La vérification de la consistance d'une ontologie ou d'une connaissance
 - Vérification des relations entre classes
 - Classification automatique des instances
 - Trouver des inconsistances dans des ontologies larges multi-auteurs

Langage de Description *AL*

35

- **Langage de Description *AL***
 - **Exemple de ABox et de TBox**

Extension du Langage *AL*

37

- **Le Langage de Description *AL* est Minimal.**
 - Le Langage *AL* a une expressivité limitée.
 - Il est possible d'augmenter l'expressivité en étendant *AL* par des operateurs. On obtient alors des langages plus expressifs
 - Convention de nommage de langages obtenu: *AL* suivi de l'abréviation des operateurs additionnels

Extension du Langage AL

38

- **Extension \mathcal{C} : Complémentation Générale $\neg C$**
 - La négation dans AL ne s'applique qu'aux concepts atomiques
 - On a souvent besoin du complément général
 - $(\neg C)^I = \Delta^I \setminus C^I$
 - $AL + \mathcal{C}$ donne le langage ALC

Extension du Langage AL

39

□ Extension \mathcal{U} : Union $C \sqcup D$

□ Permet l'union de deux concepts complexes

□ $(C \sqcup D)^I = C^I \cup D^I$

□ $AL + \mathcal{U}$ donne le langage $AL\mathcal{U}$

□ Extension \mathcal{E} : quantification existentielle pleine $\exists R \bullet C$

□ Tout concept peut être utilisé contrairement à la quantification existentielle limitée

□ $(\exists R \bullet C)^I = \{a \in \Delta^I \mid \exists b ((a, b) \in R^I \wedge b \in C^I)\}$

Extension du Langage *AL*

40

□ Propriétés

- $C \cup D \equiv \neg (\neg C \sqcap \neg D)$

- $(\exists R \bullet C) \equiv \neg (\forall R \bullet \neg C)$

- Ainsi l'Union et la quantification existentielle complète peuvent être modélisées par la complémentation générale.

- Ainsi *ALC* et *ALME* ont le même pouvoir expressif.

- Par conséquent *ALC* sera utilisé pour référer aussi *ALME*

Extension du Langage \mathcal{AL}

41

□ Extension \mathcal{N} : Restriction en nombre non qualifiée $\geq n R$ et $\leq n R$

□ $\geq n R$: au moins n

□ $\leq n R$: au plus n

□ $\geq m R \sqcap \leq n R$: individus en relation R avec au moins m individus et au plus avec n individus .

□ $(\geq n R)^I = \{a \in \Delta^I \mid \|\{b \mid (a, b) \in R^I\}\| \geq n\}$

□ $(\leq n R)^I = \{a \in \Delta^I \mid \|\{b \mid (a, b) \in R^I\}\| \leq n\}$

□

Extension du Langage *AL*

42

- Equivalences terminologiques **définissantes** (définitions).
 - ▣ Une Equivalence terminologique est définissantes si elle est acyclique après simplification de l'expression
 - $C \equiv D \sqcap E$ (acyclique, définissant)
 - $C \equiv D \sqcup \exists R \bullet E$ (cyclique, non définissant)
 - $C \equiv D \sqcup \exists R \bullet (C \sqcap \neg C)$ (cyclique, définissant)
 - après simplification $C \equiv D \sqcup ((\exists R \bullet \perp) \equiv D)$
 - ▣ TBox acyclique: une TBox dans laquelle toutes les définitions sont définissants

Extension du Langage *AL*

43

- **TBox cyclique:**
 - ▣ une TBox qui contient des définitions récursives de concepts
 - ▣ La cyclicité augmente souvent la complexité de la TBox
 - ▣ **Sémantique du Point Fixe** pour traiter des définitions récursives

Langage de Description *AL*

45

- **Langage de Description *AL***
 - **Exemple de ABox et de TBox**

Inférences en LD

□ Inférences de complétion:

□ **héritage** (propriétés héritées)

□ **combinaison** : création de descriptions par combinaison (opérateurs) de concepts.

■ Parent = Mère \sqcup Père = ...

= Personne \sqcap \exists aEnfant • Personne

□ **propagation** : Quand une assertion est faite à propos d'un individu, des conséquences logiques pour d'autres individus peuvent être propagées à l'aide de règles:

■ {aEnfant (marie, jeanne)} \cup {aEnfant (jeanne, claire)} \Rightarrow Grand-mère (marie)

Inférences en LD

□ Inférences de complétion:

- **détection de contradiction** : faits contradictoires affirmés à propos d'un individu.
 - Femme(a) ; Père(a)
- **détection de concepts incohérents** : combinaisons de restrictions sur un concept qui le rendent impossible.
 - **Satisfaction** : un concept C peut être satisfait relativement à une terminologie \mathcal{T} , s'il existe un modèle I de \mathcal{T} tel que C^I est non vide. I est alors un modèle de C.
 - $\Sigma \not\models C \equiv \perp$
 - $C = \text{Père} \sqcap \text{Mère} = \dots = \text{Personne} \sqcap \exists a\text{Enfant} \bullet \text{Personne} \sqcap (\text{Femelle} \sqcap \neg\text{Femelle}) = \text{Personne} \sqcap \exists a\text{Enfant} \bullet \text{Personne} \sqcap \perp = \perp$

Inférences en LD

□ Inférences de complétion:

- **Satisfaction Générale** : Est-ce que le système entier (Σ) est satisfaisable?

- $\Sigma \not\models \perp$ est ce que le système a un modèle?

- **Satisfaction** : un concept C peut être satisfait relativement à une terminologie \mathcal{T} , s'il existe un modèle I de \mathcal{T} tel que C^I est non vide. I est alors un modèle de C.

- $\Sigma \not\models C \equiv \perp$

- $C = \text{Père} \sqcap \text{Mère} = \dots = \text{Personne} \sqcap \exists \text{aEnfant} \bullet \text{Personne} \sqcap (\text{Femelle} \sqcap \neg \text{Femelle}) = \text{Personne} \sqcap \exists \text{aEnfant} \bullet \text{Personne} \sqcap \perp = \perp$

Inférences en LD

□ Inférences de classement et de subsomption:

□ **classement de concept** : trouver tous les concepts plus généraux/plus spécifiques qu'un concept C.

■ Concept Mère ;

■ Concepts plus généraux = {Personne, Femelle et Femme, ... };

■ Concepts plus spécifique = {Grand-mère et Mère-féconde, ... }

□ **subsomption** : le concept D est-il plus général que C ? (les individus qui satisfont C satisfont-ils forcément D ?)

■ Un concept C est subsumé par un concept D relativement à une terminologie \mathcal{T} si $C^I \subseteq D^I$ pour tout modèle I de \mathcal{T} .

■ On note alors : $C \sqsubseteq_{\mathcal{T}} D$ ou $\mathcal{T} \models C \sqsubseteq D$

Inférences en LD

□ Inférences de classement et de subsomption:

□ **Test d'instance** : l'individu a est-il instance du concept C ?

■ $\Sigma \models C(a)$?

■ **classement d'individus** : trouver tous les concepts satisfaits par un individu

■ $\{ C \mid \Sigma \models C(a) \}$

■ **Recherche d'instances**: trouver toutes les instances d'un concept C

■ $\{ a \mid \Sigma \models C(a) \}$

Inférences en LD

□ **Équivalence :**

- Deux concepts C et D sont équivalents relativement à une terminologie \mathcal{T} si : $C^I = D^I$ pour tout modèle I de \mathcal{T} .
- On note : $C \equiv_{\mathcal{T}} D$ ou $\mathcal{T} \models C \equiv D$

□ **Disjonction :**

- Deux concepts C et D sont disjoints relativement à une terminologie \mathcal{T} si : $C^I \cap D^I = \emptyset$ pour tout modèle I de \mathcal{T} .

□ Utilisation:

- recherche de tous les concepts satisfaits par un individu nouveau,
- ajout de nouveaux liens correspondants aux liens de subsumptions découverts,
- placement automatique des nouveaux individus dans la taxonomie
- Etc.

Inférences en LD

□ Réduction à la Subsumption:

□ Soient C et D des concepts:

□ C ne peut pas être satisfait \Leftrightarrow C est **subsumé** par \perp ($C \subseteq \perp$)

□ C et D sont **équivalents** \Leftrightarrow C est **subsumé** par D et D est **subsumé** par C ;

■ $(C \equiv D) \Leftrightarrow (C \subseteq D \wedge D \subseteq C)$

□ C et D sont **disjoints** $\Leftrightarrow C \sqcap D$ est **subsumé** par \perp

■ $(C \sqcap D) \subseteq \perp$

Inférences en LD

□ Réduction à la Satisfaction

- C est **subsumé** par D $\Leftrightarrow C \sqcap \neg D$ ne peut pas être satisfait
- C et D sont **équivalents** $\Leftrightarrow (C \sqcap \neg D)$ et $(\neg C \sqcap D)$ ne peuvent pas être satisfaits ;
- C et D sont **disjoints** $\Leftrightarrow C \sqcap D$ ne peut pas être satisfait

□ Réduction de la non satisfaction :

- C ne peut pas être satisfait $\Leftrightarrow C$ est subsumé par \perp
 $\Leftrightarrow C$ et \perp sont équivalents $\Leftrightarrow C$ et T sont disjoints
- C ne peut être s

Inferences en LD

□ **Consistance du ABox :**

- Une ABox \mathcal{A} est consistante relativement à une TBox \mathcal{T} , s'il existe une interprétation I qui est à la fois modèle de \mathcal{T} et de \mathcal{A} .
- \mathcal{A} est consistant s'il est consistant relativement à la TBox vide

Terminologie

□ Définitions:

- *Définitions*. Égalité dans laquelle le membre gauche est un concept atomique. Une égalité attribue des noms symboliques aux descriptions complexes. $M\grave{e}re \equiv Femme \sqcap \exists aEnfant.Personne$
- *Terminologie* \mathcal{T} : ensemble fini de définitions dans lequel il n'y a pas de définitions multiples d'un même nom de concept.

Terminologie

□ **Axiomes terminologiques :**

□ **Inclusion : $C \subseteq D, (R \subseteq S),$**

- C est inclus dans D; C est un sous-concept de D; tout individu de C est aussi un individu de D; C est subsumé par D
- I est un modèle de $C \subseteq D$ si $C^I \subseteq D^I$

- C, D: Concepts ; R, S : Rôles

□ **Equivalence : $C \equiv D, (R \equiv S),$**

- Chaque individu de C est également individu de D et vice-versa
- $C^I = D^I$
- Utilisé pour définir de nouveaux Concepts

Terminologie

- **Interprétation de base d'une terminologie \mathcal{T} :** interprétation qui n'interprète que les symboles de base de \mathcal{T} .
- **Terminologie contenant des axiomes d'inclusion:**
 - définition (incomplète) de certains concepts sous forme de spécialisation (inclusion dont le membre gauche est atomique)

Femme \sqsubseteq Personne

Sémantique du Abox

- **Sémantique du Abox.** définie en étendant les interprétations aux noms d'individus. associe à chaque nom d'individu a un élément $a^I \in \Delta^I$.

- Une interprétation I satisfait :
 - $C(a)$ si $a^I \in C^I$
 - $R(a,b)$ si $(a^I, b^I) \in R^I$.
 - un ABox \mathcal{A} , s'il satisfait toutes les assertions dans \mathcal{A} . I est un modèle de \mathcal{A} .

- I satisfait une assertion α (ou un ABox \mathcal{A}) relativement à une TBox \mathcal{T} :
 - I est à la fois modèle de α (ou de \mathcal{A}) et de \mathcal{T} .

Complexité de *ALC*

60

Langage	$\models C \subseteq D$	$\models C(a)$
<i>AL</i>	P	P
<i>ALE</i>	P	PSPACE
<i>ALC</i>	PSPACE	PSPACE
<i>ALC</i> Tbox cyclique	ExpTIME	ExpTIME

Extension sur les Rôles

62

□ Modélisation de Rôles complexes

- Introduits dans une RBox
- Langage de base : *ALCQE*

□ Extension Transitivité des rôles R^+ :

- Contient implicitement la fermeture transitive complète des faits du Abox
- $(R^+)^I = \bigcup_{i \geq 1} (R^+)^I$
- *ALCQE* + R^+ = δ

Extension sur les Rôles

63

□ Extension : Transitivité des rôles R^+ :

□ Exemple

□ Abox:

- $\text{parent}(\text{pierre}, \text{jean}), \text{parent}(\text{marie}, \text{jean}),$
- $\text{parent}(\text{jacques}, \text{pierre}), \text{parent}(\text{anne}, \text{pierre}),$
- $\text{parent}(\text{emile}, \text{marie}), \text{parent}(\text{amelie}, \text{marie}),$

□ Rbox

- $\text{ancetre} = \text{parent}^+$

□ Tbox

- $C_ANCETRE \equiv \exists \text{ ancetre}$

Extension de

64

- **Extension de δ : Hiérarchie des rôles : \neq : $R \subseteq S$**
 - $(R \subseteq S)^I = R^I \subseteq S^I$
 - Ajoute des restrictions supplémentaires sur les relations dans la Abox et augmente les possibilités de raisonnement:
 - Consistance du Abox, subsumption des rôles, appartenance aux rôles, ...
 - La construction de rôles complexes n'est pas supportée par \neq
 - Exemple
 - $\text{dirige} \subseteq \text{estMembreDe}^2$

Extension de

65

□ Autres Extensions de δ :

□ Intersection de rôles

- $(R \sqcap S)^I = R^I \cap S^I$

□ Union de rôles

- $(R \sqcup S)^I = R^I \cup S^I$

□ Négation d'un rôle

- $(\bar{R})^I = \Delta^I \times \Delta^I \setminus R^I$

□ Composition de rôles

- $(R \circ S)^I = \{ (a, c) \in \Delta^I \times \Delta^I \mid \exists b (a, b) \in R^I \wedge (b, c) \in S^I \}$

□ Exemple

- Rôles atomiques: `suitCours`, `mere`, `parent`

- Rôles complexes: `zappeCours = suitCours`; `grandMere = mere \circ parent`

Extension de

66

□ Autres Extensions de δ :

□ Inverse de rôles: R^- (?)

- Obtenu en inversant l'ordre des argument du rôle.
- $(R^-)^I = \{(b, a) \in \Delta^I \times \Delta^I \mid (a, b) \in R^I\}$

□ Exemple

- `suitCours (franck, logiqDescr),`
- Role complexe : `estSuiviPar = suitCours-`
- `Entraine : estSuiviPar(logiqDescr, franck)`

Extension de

67

□ Autres Extensions de δ :

□ Rôle Fonctionnel : $\leq 1R$ (\mathcal{F})

- La fonctionnalité restreint le nombre maximal de relation de la source à 1.
- Le rôle devient une fonction.
- Le second argument du rôle dépend fonctionnellement du premier
- $R(x, y) \Leftrightarrow y = f(x)$
- $(R^{-1})^I = \{(b, a) \in \Delta^I \times \Delta^I \mid (a, b) \in R^I\}$

□ Exemple

- Les rôles **age** et **mère** sont fonctionnels

Extension de

68

□ Autres Extensions de δ :

□ Expansion : Nominal (\emptyset)

- Permet de définir un concept en énumérant ses membres

□ Exemple

- $\text{DiplomeIAI} = \{\text{Licence}, \text{AP}, \text{MIAGE}, \text{MASTER}, \text{ING}\}$
- $\text{DiplomeIAI}^! = \{\text{Licence}^!, \text{AP}^!, \text{MIAGE}^!, \text{MASTER}^!, \text{ING}^!\}$

Principaux Langages DL

70

Abréviations	Caractéristiques
\mathcal{AL}	Langage de Base: Attribute Language
\mathcal{C}	Complément General : $\neg C$
\mathcal{U}	Union : $C \sqcup D$
\mathcal{E}	Quantification existentielle complète: $\exists R \bullet C$
\mathcal{N}	Restriction en Nombre non qualifiée: $\geq n R$ $\leq n R$
\mathcal{S}	ALC + Rôles transitifs R^+
\mathcal{H}	Hiérarchies de rôles : $R \sqsubseteq S$
\mathcal{I}	Inverse de rôles : R^-
\mathcal{F}	Fonctionnel : $\leq 1 R$
\mathcal{O}	Nominal
\mathcal{R}	Inclusion de rôles complexes: $R \circ S \sqsubseteq S$; $R \circ S \sqsubseteq R$
\mathcal{Q}	Restriction en Nombre qualifiée: $\geq n R \bullet C$ $\leq n R \bullet C$

Complexité des Langages

71

Langage		
AL	P-comp	P-comp
ALE	NP-comp	PSPACE-comp
ALC	PSPACE-comp	PSPACE-comp
ALC Tbox cyclique	EXPTIME-comp	EXPTIME-comp
δ	EXPTIME-comp	EXPTIME-comp
$\delta_{\#}$	EXPTIME-comp	EXPTIME-comp
$\delta_{\#}^*$	NEXPTIME-comp	NEXPTIME-comp

Ontologies

73

□ Définition

- En philosophie : Etude de l'être en tant qu'être.
- En Informatique: *spécification formelle et explicite d'une conceptualisation partagée*

□ Buts :

- Fournir le « **sens** » des symboles utilisés pour construire un modèle du monde
- Faire en sorte que les personnes et les machines se comprennent (*partage et réutilisation de l'information*)
- *Permettre de déclencher des inférences, de faire des déductions*

Ontologies

74

Que met on dans une ontologie?

- Une Ontologie inclut généralement:
 - une organisation hiérarchique des concepts pertinents (principes, idées, catégorie d'objet, notions potentiellement abstraites)
 - des relations qui existent entre ces concepts
 - ainsi que des règles et axiomes qui les contraignent.

- **importance de l'organisation taxinomique (hiérarchie de concepts)**
 - la classification ou identification (le fait de déterminer si quelque chose appartient à une classe) et la catégorisation (le fait d'identifier les catégories existantes) sont des inférences élémentaires que nous faisons à longueur de journée.

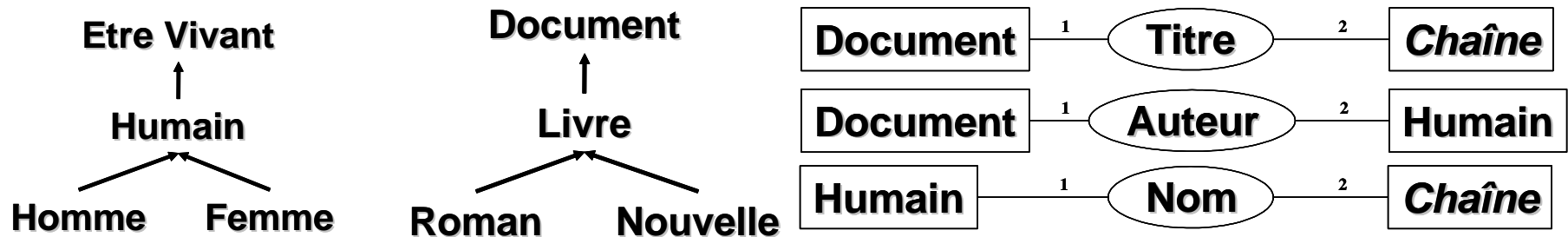
Ontologies

75

Exemple 1: Conversation entre deux personnes:

- Tu connais un *restaurant* proche ?
- Il y a une *cafeteria* au coin de la rue.
- Merci."

Exemple 2 : Recherche des livres écrits par Hugo.



Ontologies

76

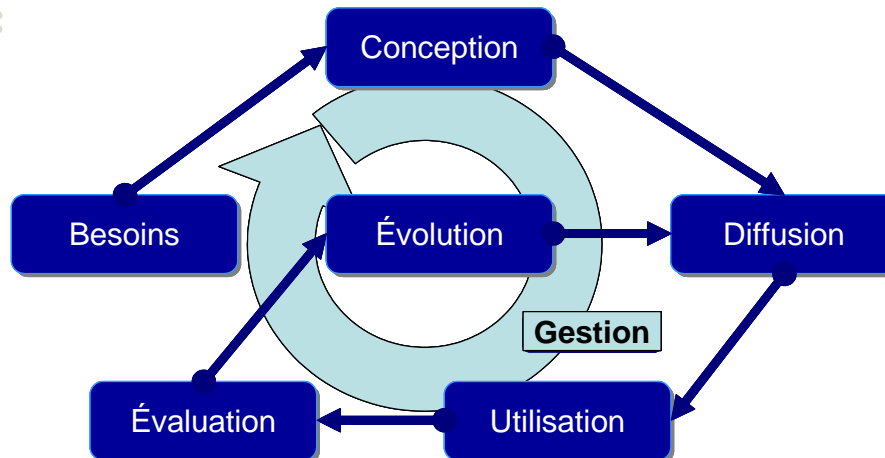
3 grands types d'ontologies

- Ontologies de **représentation des connaissances**: ontologies de **haut niveau** décrivant les notions utilisées dans les autres ontologies
 - Exemple: Dublin Core, GFO, OpenCyc, SUMO, DOLCE
- Ontologies de **domaine**: permettent de spécifier les connaissances d'un domaine de façon indépendante du type de manipulation qui vont être opérées
- Ontologies de **raisonnement**: décrit les processus de raisonnement appliqués aux connaissances

Critères d'évaluation d'une ontologie

- ❑ **Clarté** : communiquer le sens des termes définis
- ❑ **Cohérence**: pas de contradiction
- ❑ **Extensibilité**: permettre l'ajout de nouveaux concepts
- ❑ **Indépendance** (par rapport à l'implémentation)

Cycle de vie:



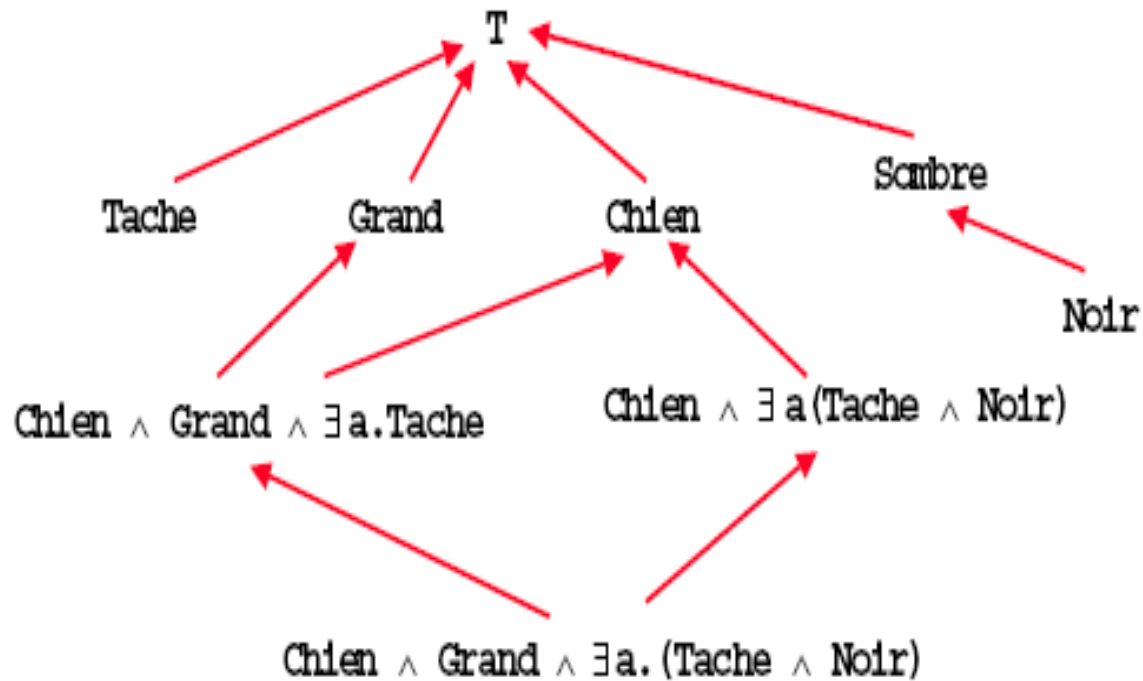
Ontologies et LD

78

- **Les LD permettent de:**
 - **formaliser** les ontologies
 - **maintenir la consistance** des ontologies
 - **produire automatiquement la hiérarchie** des concepts (via le mécanisme de subsomption) à partir des descriptions de concepts.
 - **Etendre** les ontologies: ajouter des concepts en les insérant automatiquement dans la taxonomie des concepts de l'ontologie.

Ontologies et LD

79



Ontologies

80

Exemples d'ontologies :

- ▣ **GALEN** (medical)
- ▣ **SNOMED**
- ▣ **Plant Ontology:** [http://www.obofoundry.org/cgi-bin/detail.cgi?id=po_anatomy]
- ▣ **Disease Ontology:** [http://www.obofoundry.org/cgi-bin/detail.cgi?id=disease_ontology]
- ▣ **Infectious disease Ontology:** [http://www.obofoundry.org/cgi-bin/detail.cgi?id=infectious_disease_ontology]

LANGUAGES DE REPRESENTATION

82

- RDF/RDF(S) (Resource Description Framework))
- OIL (Ontology Inference Layer)
- DAML+OIL
- OWL (Web Ontology Language)
 - ▣ OWL Lite, OWL DL, OWL Full

□ **RDF: Resource Description Framework**

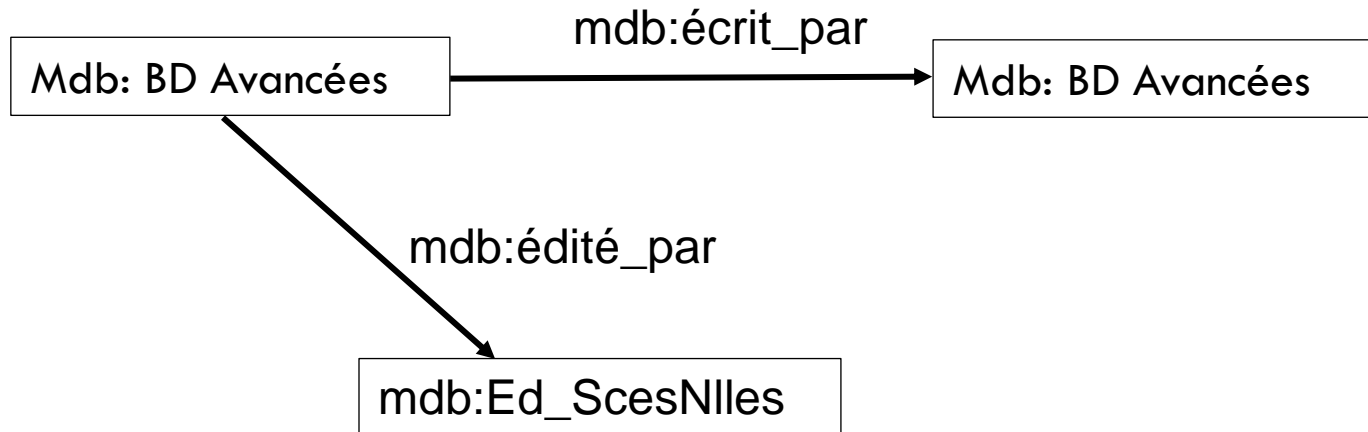
- Langage de Markup pour encoder la connaissance
- Une connaissance = un triplet
 - Un triplet = une assertion : sujet, prédicat, objet
 - Chaque élément du triplet est soit un littéral soit une ressource
 - Chaque ressource est donnée par un référence URI
 - Chaque URI représente une entité ou un concept
- RDF peut être utilisé pour définir une ontologie
 - Exemple:
 - Mdb: BD Avancées mdb:écrit_par mdb:Jean_Gérard
 - Mdb: BD Avancées mdb:édité_par mdb:Ed_ScesNlles

RDF

84

□ RDF: Resource Description Framework

- Un document RDF peut-être vu comme un graphe étiqueté



RDF/S

85

□ RDF/S

- **RDF** n'impose pas de restriction particulières sur les étiquettes
- Pour fixer les étiquettes utilisables, on utilise **RDF/S** pour fixer les ressources valides
- RDF/S fixe le vocabulaire utilisé dans les graphes RDF:
 - Définit les classes et propriétés utilisées
 - Introduit des relations (prédicats) prédéfinies entre les classes/instances:
 - sub-classe
 - sub-property
 - range
 - domains
 - Etc

SEMANTIQUE DE RDF

86

- **La sémantique de RDF peut être établie selon deux approches**
 - Approche directe théorie du modèle
 - En transcrivant les assertions RDF en logique
 - Peut conduire à des expressions logiques d'ordre supérieur si on ne met pas de restriction
 - Les restrictions permettent de réduire la complexité

SPARQL

87

□ SPARQL Protocol And RDF Query Language

□ Un langage de requête et un protocole pour accéder à RDF

- Permet d'extraire des informations sous forme d'URIs
- Extraire des sous-graphes
- Construire de nouveaux graphes RDF à partir des informations extraites

□ Principe

- Définir des patterns de triplets ou certains items peuvent être des variables
- Les patterns vont ensuite être matchés avec le graphe
 - **?livre dc:titre ?titre**
- Utilise un langage style SQL (select ... from ... where)

SPARQL

88

□ Exemple

- SELECT ?title
 - FROM http://example.org/library
 - WHERE { <http://example.org/book/book1> dc:title ?title }
-
- SELECT donne la liste des valeurs à retourner
 - FROM identifie le nome du graphe RDF
 - WHERE Les patterns sous forme d'une liste de triplets
 - + **LIMIT, ORDER BY, ...**

- Lien: <http://www.w3.org/TR/rdf-sparql-query/>

DAML+OIL

90

- De base, le web sémantique se traduit sous la forme d'assertions RDF
- Comment utiliser efficacement ces assertions pour déduire de nouvelles connaissances?
- 2 initiatives concomitantes pour l'utilisation des DL dans le cadre du web sémantique ont été développées
 - DAML et OIL
 - Combiné en DAML+OIL

- **DAML : DARPA Agent Markup Language**
 - Langage de balisage pour agent basé sur RDF
 - De 1999 à 2006, Jim Hendler (University of Maryland)
 - Met l'accent sur des outil facilitant l'expression intuitive des connaissances

□ OIL : Ontology Inference Layer

- Initiative universitaire pour définir une fondation logique pour le web-sémantique :
 - Dieter Fensel, Frank van Harmelen (VU Amsterdam)
 - Ian Horrocks (University of Manchester)
 - Deborah McGuinness (Stanford), etc.
- Une proposition pour une représentation basée-web et une couche d'inférence pour les ontologies
- Fournit un ensemble de primitives de modélisation largement utilisées dans les langage de frames (objet)
- + la sémantique formelle et les services de raisonnement apportés par les Logiques de Description
- Compatible avec RDF Schema avec une sémantique précise pour définir la signification des termes

□ OIL : Ontology Inference Layer

- Suit une approche par couche pour la définition d'un langage de web-sémantique
- Chaque couche ajoute des fonctionnalités et par ricochet augmente la complexité
 - CORE OIL coïncide pour une large part avec RDF (à quelques exceptions près)
 - STANDARD OIL : Inclut la plupart des primitives de modélisation nécessaires avec une sémantique claire, un pouvoir expressif et un mécanisme d'inférence adéquats
 - AUTRES COUCHES
 - INSTANCE OIL
 - Heavy OIL

