Logiques de Description

XML, un langage d'arbre

Master Recherche SIA - Master Pro ILI

Année 2008-09

Logiques de Description

Objectifs de la représentation des connaissances :

- fournir un moyen de décrire le monde
- et les outils pour raisonner sur ces connaissances

Historiquement, deux approches (années 70) :

- formalismes basés sur la logique (essentiellement du premier ordre)
- approches ad hoc, basées sur des structures de graphes

Les approches basées sur la logique

Intérêts

- approche généraliste
- le langage de description est généralement une variante de la logique du premier ordre
- raisonner sur un tel langage revient à tester ou à calculer des conséquences logiques

Inconvénients

- éloigné des domaines d'application
- difficulté d'adaptation pour modéliser les connaissances d'un point de vue "humain"

Les approches basées sur des structures de réseaux

Les approches non-basées sur la logique ont eu plus de succès :

- de nombreux langages ont été proposés (avec des formalismes pensés pour des domaines d'application spécifiques)
- avec des procédures ad hoc pour effectuer le raisonnement

Deux cadres différents :

- les réseaux sémantiques proposés par Quillian (1967)
- frame systems proposés par Minsky (1981)

Tous les deux peuvent être vus

- comme des structures de réseaux
- où la structure représente des ensembles d'individus et leurs relations.

Limites des approches basées sur des structures réseaux

- manque de caractérisation sémantique précise
- comportements parfois différents pour le raisonnement

Les premiers travaux qui ont donné une sémantique formelle pour les principales caractéristiques ont montré

- qu'elles sont basées sur un fragment de la logique du premier ordre
- ajout de caractéristiques spécifiques $\Rightarrow$ différents fragments
- $\Longrightarrow$ différentes classes de problèmes et différentes classes de complexité pour les algorithmes de raisonnement

Logiques de Description : fruit de l'évolution des travaux

- pour donner une sémantique formelle
- et mieux classifier les précédentes approches.

Les logiques de description

Éléments fondamentaux :

- des individus (*Marie, Mehdi, Jonathan, Mounir*)
- des concepts : des ensembles d'individus (*master2ILI*, *master2SIA*, *unitesEnseignement*)
- des rôles : des relations entre des concepts (*suitUnCours*, *estUn*)

Services

Les principales tâches d'inférence sont

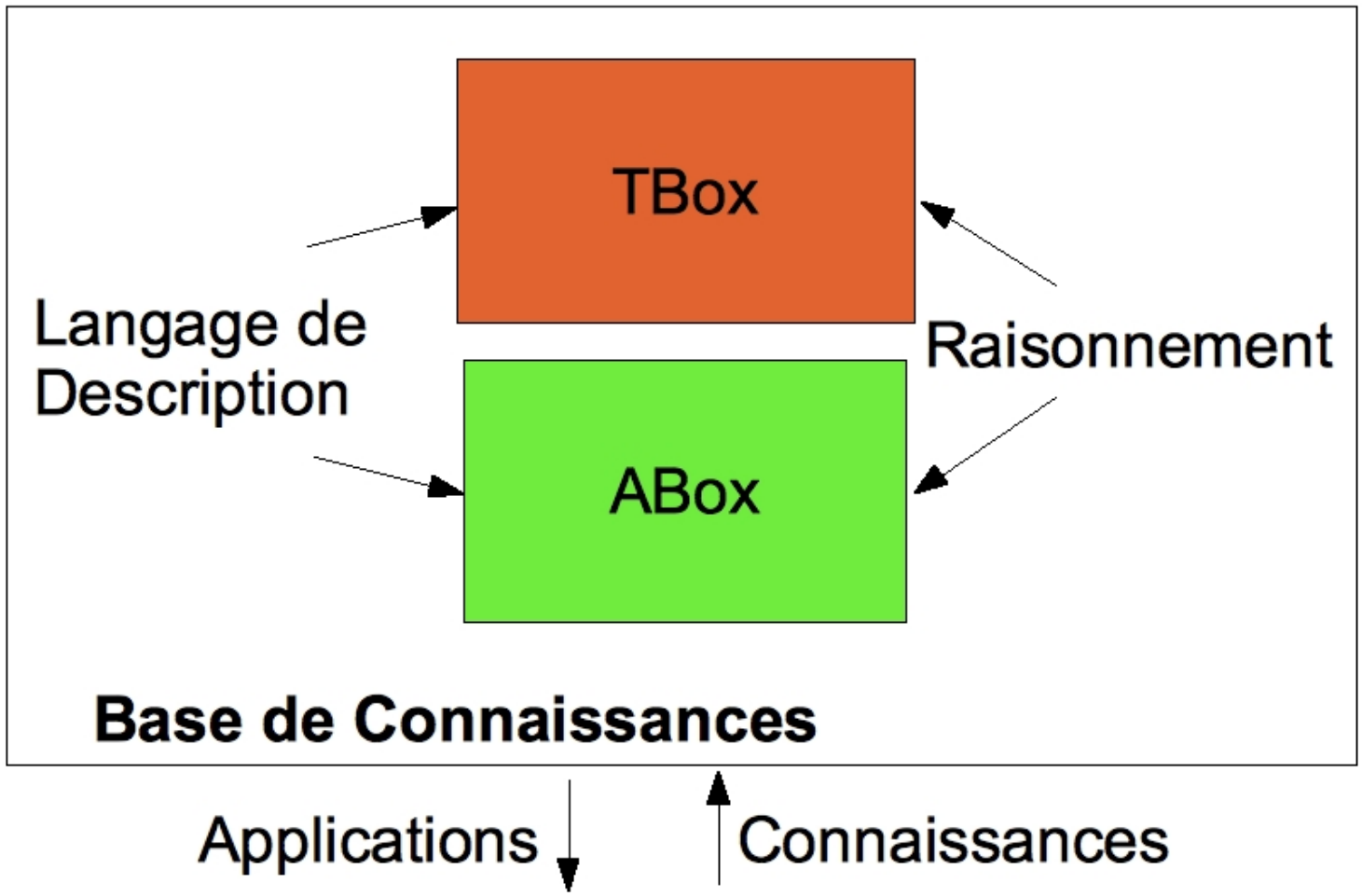
- la satisfiabilité : la description comporte-t-elle une contradiction ?
- la subsomption : concept est-il plus général qu'un autre ?
- l'appartenance : un individu appartient-il à un certain concept ?
- la cohérence : existe-t-il un modèle pour cette description ?
- l'énumération des éléments : quelles sont les instances d'un concept ?

TBox, ABox

Les logiques de description reposent sur une division de la représentation des connaissances et des tâches d'inférence :

- la TBox : la *Terminology Box*, ou *Taxonomy Box* contient la modélisation du monde en termes de concepts, de leurs propriétés, et des relations entre les concepts. C'est la connaissance en *intention* du monde. Ses tâches de raisonnement sont principalement la satsfiabilité et la subsomption ;
- la ABox : la *Assertional Box* contient la connaissance qui concerne les individus. C'est la connaissance en *extension*. Les tâches de raisonnement sont l'appartenance, la cohérence, l'énumération des instances.

TBox, ABox


TBox
Langage de Description
Raisonnement
ABox
Base de Connaissances
Applications
Connaissances

TBox : Langage de Description

- concepts atomiques et rôles atomiques (primitives)
- définition de concepts de concepts complexes à partir de ces primitives et d'opérateurs

Les différents types d'opérateurs permettent de distinguer les différentes familles de langage.

$\mathcal{AL}$-Langage : langage d'attributs (Schmidt-Schauss et Smolka, 1991) : langage minimal. Les autres langages de cette famille sont construits par extension.

$\mathcal{AL}$-Langage

Opérateurs de construction de concepts :

- les concepts atomiques A, B
- le concept universel $\top$
- le concept bottom $\bot$
- la négation atomique $\neg A$
- l'intersection de concepts $C \sqcap D$
- la restriction de valeurs $\forall R.C$
- la quantification existentielle limitée $\exists R.\top$

$\mathcal{AL}$-Langage - Un exemple

Concepts atomiques :

$\texttt{Employé}, \texttt{ChefService}, \texttt{ProjetDvpt}, \texttt{ProjetAnalyse}$

Rôles atomiques :

$\texttt{travailleSur}, \texttt{aPourChef}$

Les employés qui ne sont pas chefs de service :
$\texttt{Employé} \sqcap \neg\texttt{ChefService}$
Les chefs de service qui travaillent sur un projet :
$\texttt{ChefService} \sqcap \exists\texttt{travailleSur}.\top$
Les employés actuellement affectés uniquement à un projet d'analyse :
$\texttt{Employé} \sqcap \forall\texttt{travailleSur}.\texttt{ProjetAnalyse}$

Sémantique formelle pour $\mathcal{AL}$

On peut donner une sémantique en logique du premier ordre à toutes ces constructions :

- $\mathcal{I}$ est une interprétation constituée
  - d'un ensemble non-vide $\Delta^{\mathcal{I}}$ (le domaine d'interprétation)
  - les concepts sont vus comme des prédicats unaires sur $\Delta^{\mathcal{I}}$: formule $A(x)$, et pour toute interprétation $\mathcal{I}$ l'ensemble des éléments satisfiant $A(x)$ est $A^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$;
  - les rôles sont vus comme des prédicats binaires sur $\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$.
- $\mathcal{I}$ est un modèle pour C si $C^{\mathcal{I}} \neq \emptyset$

Sémantique formelle pour $\mathcal{AL}$

DL	**FOL**	**Ensembliste**
A	$A(x)$	$A^{\mathcal{I}}$
$\bot$	`false`	$\emptyset$
$\top$	`true`	$\Delta^{\mathcal{I}}$
$\neg A$	$\neg A(x)$	$\Delta^{\mathcal{I}} / A^{\mathcal{I}}$
$C \sqcap D$	$C(x) \wedge D(x)$	$C^{\mathcal{I}} \cap D^{\mathcal{I}}$
$\forall R.C$	$\forall x.R(y,x) \rightarrow C(x)$	$\{a \in \Delta^{\mathcal{I}} / \forall b.(a,b) \in R^{\mathcal{I}} \rightarrow b \in C^{\mathcal{I}}\}$
$\exists R.\top$	$\exists x.R(y,x)$	$\{a \in \Delta^{\mathcal{I}} / \exists b.(a,b) \in R^{\mathcal{I}}\}$

Opérateurs supplémentaires

On obtient des langages plus expressifs si on ajoute de nouveaux opérateurs :

- l'union de concepts $\mathcal{U}$: $C \sqcup D$
- la quantification existentielle complète $\mathcal{E}$: $\exists R.C$
- la négation généralisée $\mathcal{C}$ (complémentaire) : $\neg C$

Opérateurs supplémentaires - Un exemple

Les chefs qui n'ont pas de chef :

$\texttt{ChefService} \sqcap \neg(\exists \texttt{aPourChef}.\top)$

Les employés actuellement affectés à un projet d'analyse (pas forcément uniquement) :

$\texttt{Employé} \sqcap \exists \texttt{travailleSur}.\texttt{ProjetAnalyse}$

Ceux qui travaillent sur un projet d'analyse ou de développement :

$\exists \texttt{travailleSur}.\texttt{ProjetAnalyse} \sqcup$
$\exists \texttt{travailleSur}.\texttt{ProjetDvpt}$

Sémantique formelle des nouveaux opérateurs

Sémantique formelle des nouveaux opérateurs

DL	**FOL**	**Ensembliste**
$\neg C$	$\neg C(x)$	$\Delta^{\mathcal{I}} / C^{\mathcal{I}}$
$C \sqcup D$	$C(x) \vee D(x)$	$C^{\mathcal{I}} \cup D^{\mathcal{I}}$
$\exists R.C$	$\exists x.R(y,x) \wedge C(x)$	$\{a \in \Delta^{\mathcal{I}} / \exists b.(a,b) \in R^{\mathcal{I}} \wedge b \in C^{\mathcal{I}}\}$

Encore un opérateur supplémentaire

Et toujours pour augmenter l'expressivité, un opérateur supplémentaire :

- les restrictions numériques $\mathcal{N}$: $\leq_n R$ ou $\geq_n R$

Un exemple : les employés qui travaillent sur au plus trois projets

$\texttt{Employé} \sqcap \leq_3 \texttt{travailleSur}$

Encore un opérateur supplémentaire

Et toujours pour augmenter l'expressivité, un opérateur supplémentaire :

- les restrictions numériques $\mathcal{N}$: $\leq_n R$ ou $\geq_n R$

Un exemple : les employés qui travaillent sur au plus trois projets
$\texttt{Employé} \sqcap \leq_3 \texttt{travailleSur}$

DL	**Ensembliste**
$\leq_n R$	$\{a \in \Delta^{\mathcal{I}} / \lvert\{b/(a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$
$\geq_n R$	$\{a \in \Delta^{\mathcal{I}} / \lvert\{b/(a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$

Encore un opérateur supplémentaire

Et toujours pour augmenter l'expressivité, un opérateur supplémentaire :

- les restrictions numériques $\mathcal{N}$: $\leq_n R$ ou $\geq_n R$

Un exemple : les employés qui travaillent sur au plus trois projets
$\texttt{Employé} \sqcap \leq_3 \texttt{travailleSur}$

DL	**Ensembliste**
$\leq_n R$	$\{a \in \Delta^{\mathcal{I}} / \lvert\{b/(a,b) \in R^{\mathcal{I}}\}\rvert \leq n\}$
$\geq_n R$	$\{a \in \Delta^{\mathcal{I}} / \lvert\{b/(a,b) \in R^{\mathcal{I}}\}\rvert \geq n\}$

DL	**FOL**
$\leq_n R$	$\exists y_1, \ldots y_{n+1}.R(x,y_1) \wedge \cdots \wedge R(x,y_{n+1}) \rightarrow \bigvee_{i<j} y_i = y_j$
$\geq_n R$	$\exists y_1, \ldots y_n.R(x,y_1) \wedge \cdots \wedge R(x,y_n) \wedge \bigwedge_{i<j} y_i \neq y_j$

Différentes familles de langages $\mathcal{AL}$

On obtient ainsi plusieurs familles de langages qui ne sont pas toutes équivalentes, en partant de $\mathcal{AL}$ et en ajoutant un ou plusieurs des opérateurs $\mathcal{C}$, $\mathcal{E}$, $\mathcal{U}$, ou $\mathcal{N}$.

Néanmoins :

- peut-on exprimer $\mathcal{U}$ à partir d'autres opérateurs ?

Différentes familles de langages $\mathcal{AL}$

On obtient ainsi plusieurs familles de langages qui ne sont pas toutes équivalentes, en partant de $\mathcal{AL}$ et en ajoutant un ou plusieurs des opérateurs $\mathcal{C}$, $\mathcal{E}$, $\mathcal{U}$, ou $\mathcal{N}$.

Néanmoins :

- peut-on exprimer $\mathcal{U}$ à partir d'autres opérateurs ?
  $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$
- peut-on exprimer $\mathcal{E}$ à partir d'autres opérateurs ?

Différentes familles de langages $\mathcal{AL}$

On obtient ainsi plusieurs familles de langages qui ne sont pas toutes équivalentes, en partant de $\mathcal{AL}$ et en ajoutant un ou plusieurs des opérateurs $\mathcal{C}$, $\mathcal{E}$, $\mathcal{U}$, ou $\mathcal{N}$.

Néanmoins :

- peut-on exprimer $\mathcal{U}$ à partir d'autres opérateurs ?
  $C \sqcup D \equiv \neg(\neg C \sqcap \neg D)$
- peut-on exprimer $\mathcal{E}$ à partir d'autres opérateurs ?
  $\exists R.C \equiv \neg(\forall R.\neg C)$
- et $\mathcal{C}$ peut s'écrire à partir d'une combinaison de $\mathcal{E}$ et $\mathcal{U}$
- donc, $\mathcal{ALC}$ est équivalent à $\mathcal{ALUE}$

Axiomes terminologiques

- Les expressions précédentes permettent de décrire des concepts complexes
- Les axiomes permettent d'exprimer les relations entre concepts et entre rôles

Basiquement, il y a deux types de relations :

- l'inclusion : $C \sqsubseteq D$ ou $R \sqsubseteq S$
  Sémantique : $C_{\mathcal{I}} \subseteq D_{\mathcal{I}}$
- l'égalité : $C \equiv D$ ou $R \equiv S$
  Sémantique : $C_{\mathcal{I}} = D_{\mathcal{I}}$

Ces relations permettent d'ordonner les concepts entre eux, d'établir une hiérarchie, une terminologie ou taxonomie.

Exemples

$$\texttt{ChefProjet} \equiv \texttt{ChefService} \sqcap \exists \texttt{travailleSur}.\top$$

$$\texttt{Analyste} \equiv \texttt{Employé} \sqcap \forall \texttt{travailleSur}.\texttt{ProjetAnalyse}$$

$$\texttt{Développeur} \equiv \texttt{Employé} \sqcap \forall \texttt{travailleSur}.\texttt{ProjetDvpt}$$

$$\texttt{Analyste-Développeur} \equiv \texttt{Analyste} \sqcap \texttt{Développeur}$$

$$\texttt{ChefService} \sqsubseteq \texttt{Employé}$$

Sémantique des axiomes d'égalité

$C \equiv D$

- Si le terme de gauche est un concept atomique, c'est une définition : permet de nommer un nouveau concept ;
- une seule définition par concept acceptée ;
- on distingue
  - les concepts qui n'apparaissent que dans les parties droites des définitions : ce sont les primitives ou concepts de base
  - les autres concepts : les concepts définis

Sémantique des axiomes d'égalité

- une terminologie $\mathcal{T}$ est dite cyclique s'il existe un concept de $\mathcal{T}$ qui s'utilise lui-même, directement ou indirectement. $\mathcal{T}$ est acyclique sinon.
- une terminologie $\mathcal{T}$ est dite définissante si toute interprétation $\mathcal{I}$ sur les primitives admet une seule extension sur les concepts de $\mathcal{T}$ qui soit un modèle pour $\mathcal{T}$

Theorem

Toute terminologie définissante de $\mathcal{ALC}$ est équivalente à une terminologie acyclique.

Sémantique des axiomes d'égalité

Pour les terminologies cycliques :
$\texttt{SupérieurHiérarchique} \equiv \texttt{ChefService} \sqcup \forall \texttt{estChefDe.SuperieurHierarchique}$

La sémantique de ces concepts est donnée par une sémantique de point fixe.

Theorem
Soient $\mathcal{T}$ une terminologie, $\mathcal{I}$ une interprétation, et $\mathcal{J}$ une restriction de $\mathcal{I}$ sur les primitives de $\mathcal{T}$. Alors $\mathcal{I}$ est un modèle de $\mathcal{T}$ si et seulement si $\mathcal{I}$ est un point fixe de $\mathcal{T}_{\mathcal{J}}$

Sémantique des axiomes d'inclusion

On se ramène au cas de l'égalité, en ajoutant un nouveau concept $\overline{C}$ pour tout concept C tel que $C \sqsubseteq D$.
Cette normalisation permet d'écrire :

$$C \equiv \overline{C} \sqcap D$$

Néanmoins, la relation d'inclusion est importante pour l'expressivité.

ABox

Cette partie permet de définir les individus et les relations entre les individus :

```
Employé(Paul).   Employé(Amina).    Employé(Ahmed).
ChefService(Amina).
ProjetDvpt(LensJazzFestival).
estChefDe(Amina,Paul).   estChefDe(Amina,Ahmed).
travailleSur(Paul,LensJazzFestival).
travailleSur(Amina,LensJazzFestival).
```

On pourra déduire que : `ChefProjet(Amina).`

Des individus dans la TBox ?

D'autres opérateurs ont été étudiés pour permettre de nommer des individus ou des propriétés dans la TBox :

- le constructeur ensembliste set permet de définir des ensembles d'individus
  $\texttt{EquipeDvpeur} \equiv \{\texttt{Paul}, \texttt{Amina}\}$
  Cet opérateur peut être obtenu avec un constructeur de singleton est $\mathcal{U}$
- l'opérateur de spécialisation $R : a$
  $\texttt{EquipeLJF} \equiv \texttt{travailleSur.LensJazzFestival}$
  Cet opérateur est équivalent à $\exists R.\{a\}$

Raisonner sur la TBox

Raisonner sur la TBox, c'est essentiellement tester la subsomption entre deux concepts :

Definition

Un concept C est subsumé par un concept D par rapport à $\mathcal{T}$ si $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$.

ce qui permet d'implémenter les autres inférences :

Theorem

Soient C et D deux concepts, alors

1. *C est insatisfiable ssi C est subsumé par $\bot$*
2. *C et D sont équivalents ssi C est subsumé par D et D est subsumé par C*
3. *C et D sont disjoints ssi $C \sqcap D$ est subsumé par $\bot$.*

Exercice

Proposer une représentation des membres d'une famille sur plusieursgénérations. On veut connaître (ou déduire) les relations entre membres de la famille :

- père, mère, frère, soeur ;
- grand-père, grand-mère, oncle, tante
- ancêtre
- a une descendance de filles
- ...