Les logiques de descriptions

Les logiques de descriptions LD sont des formalismes de représentation des connaissances descendant du système **KL-ONE** (Brachman 2003) qui lui-même est issu à la fois des **réseaux sémantiques** (Ross Quillian 1968) et des **frames**(M. .Minsky 1975).

Un réseau KL-ONE est un ensemble d'éléments en relation les uns avec les autres.

Les éléments du réseau sont des ***concepts*** et les relations du réseau sont appelées des ***rôles***.

Les concepts peuvent être *primitifs* (ils possèdent alors des propriétés nécessaires mais pas suffisantes) ou *définis* (ils possèdent alors des propriétés nécessaires et suffisantes).

Si C est un concept défini, le système KL-ONE peut répondre à des questions du type : '' *le concept C* ***subsume****-t-il le concept D*? ou *le concept C est-il subsumé par le concept D*?

KL-ONE est doté d'un ***classifieur*** qui place les nouveaux concepts définis dans la hiérarchie existante

Les LD utilisent une approche ontologique : pour décrire les individus d'un domaine, elles requièrent tout d'abord la définition des catégories générales d'individus et les relations logiques que les individus ou catégories peuvent entretenir entre eux. Comme leur nom l'indique, elles sont équipées d'une sémantique formelle, et un accent est mis sur le raisonnement. Cela permet d'inférer des connaissances implicites à partir des connaissances explicites contenus dans la BC (intelligence des applications à base de DL).

Les DL sont des formalismes pour représenter des connaissances, et comme dans les représentations des connaissances on cherche tjs à proposer des systèmes qui doivent répondre à une interrogation d'un utilisateur en un temps raisonnable, beaucoup de chercheurs se sont intéressés aux procédures de décision qui doivent tjs se terminer pour les réponses positives ou négatives. La décidabilité et la complexité du pb de l'inférence dépend de la puissance d'expressivité de la logique de description proposée. Plus elle est expressive plus la complexité et l'indécidabilité augmente et inversement. La puissance de l'expressivité doit donc être un compromis.

Les premiers travaux sur les LD commencèrent au début des années 1980 avec des systèmes à base de connaissances LOOM, CLASSIC. Ces premières implémentations résolvent des problèmes d'inférence (détecter les relations implicites, par exemple) en temps souvent polynomial, mais ne s'appliquent qu'à des LD peu expressives

Les LD expressives ou très expressives, même si elles sont en NP, sont très utilisées ces derniers temps notamment dans les nouvelles applications telles que le web sémantique. En pratique, le comportement des algorithmes est souvent acceptable.

La modélisation des connaissances d'un domaine avec les LD se réalise en deux niveaux. Le premier, le niveau terminologique ou Tbox, décrit les connaissances générales d'un domaine alors que le second, le niveau factuel ou Abox, représente une configuration précise

Une Tbox comprend la définition des concepts et des rôles,

alors qu'une Abox décrit les individus en les nommant et en spécifiant en terme de concepts et de rôles des assertions qui portent sur ces individus nommés.

Plusieurs Abox peuvent être associés à une même Tbox; chacune représente une configuration constituée d'individus, et utilise les concepts et rôles de la Tbox pour l'exprimer.

Langage *AL*(attributive language) c'est le langage minimal

Les logiques de descriptions disposent de deux composantes : la ***T-Box*** et la ***A-Box*** qui ont chacune leur propre langage et leurs mécanismes d'inférences associés.

Le langage terminologique de la T-box permet de définir des concepts et le langage assertionnel de la A-box permet de décrire les faits ou les règles incidentes liées aux concepts

Les règles syntaxiques sont définies inductivement à partir d'un ensemble P de concepts primitifs et d'un ensemble R de rôles :

$P \rightarrow C, D/ \top / \bot / C \wedge D/ \neg C/ \forall rC/ \exists r \top$ / …

C, D : concepts primitifs, $\top$: concept général, $\bot$: concept spécifique,

r : rôle

Les différents systèmes basés DL diffèrent par la description de leurs langages. Ils ont une sémantique basée sur la théorie des modèles de la LPO. Les formules dans les Tbox et Abox sont des formules du premier ordre ou une légère extension de ces formules.

Les langages assertionnels et les A-Box

On trouve deux types principaux d'assertions :

- Les assertions d'appartenance à un concept

Ex:

Homme(Jean) pour Jean est un homme

- Les assertions d'appartenance à un rôle

Ex :

relParentEnfant(Sophie,Anne) pour Anne est la fille deSophie

sémantique

Pour la sémantique formelle de ***AL***-concepts, on considère une interprétation $\boldsymbol{I}$ qui consiste en un ensemble non vide (le domaine d'interprétation) $\Delta^{\boldsymbol{I}}$ et une fonction d'interprétation qui assigne à chaque concept atomique A un ensemble $A^{\boldsymbol{I}} \subseteq \Delta^{\boldsymbol{I}}$ et à chaque rôle R une relation binaire $R^{\boldsymbol{I}} \subseteq \Delta^{\boldsymbol{I}} \times \Delta^{\boldsymbol{I}}$. La fonction d'interprétation est étendu à la description du concept par les définitions inductives :

$\top^{\boldsymbol{I}} = \Delta^{\boldsymbol{I}}$

$\bot^{\boldsymbol{I}} = \emptyset$

$(\neg A)^{\boldsymbol{I}} = \Delta^{\boldsymbol{I}} \setminus A^{\boldsymbol{I}}$

$(C \wedge D)^{\boldsymbol{I}} = C^{\boldsymbol{I}} \cap D^{\boldsymbol{I}}$

$(\forall rC)^{\boldsymbol{I}} = \{\mathbf{a} \in \Delta^{\boldsymbol{I}} \mid \forall b.(a, b) \in R^{\boldsymbol{I}} \rightarrow b \in C^{\boldsymbol{I}}\}$

$(\exists r \top)^{\boldsymbol{I}} = = \{\mathbf{a} \in \Delta^{\boldsymbol{I}} \mid \exists b.(a, b) \in R^{\boldsymbol{I}}$

Deux concepts C,D sont équivalent notée $C \equiv D$ si $C^{\boldsymbol{I}} = D^{\boldsymbol{I}}$ pour toutes les interprétations $\boldsymbol{I}$ **.** Par exemple, on peut facilement vérifier que

$\forall$aenfantf.Femelle $\wedge$ $\forall$aenfant.Etudiant et $\forall$ aenfant(Femelle $\wedge$ Etudiant) sont éq.

TBox	ABox
Femelle $\subseteq \top \wedge \neg$Mâle Mâle $\subseteq \top \wedge \neg$Femelle Animal $\equiv$ Mâle $\vee$Femelle Humain $\subseteq$Animal Femme $\equiv$ Humain $\wedge$ Femelle Homme $\equiv$ Humain $\wedge \neg$Femelle Mère $\equiv$ Femme $\wedge \exists$relParentEnfant Père $\equiv$ Homme $\wedge \exists$relParentEnfant MèreSansFille $\equiv$ Mère $\wedge$ $\forall$relParentEnfant.. $\neg$Femme relParentEnfant$\subseteq \top_R$	Humain(Anne) Femelle(Anne) Femme(Sophie) Humaine(Robert) $\neg$Femelle(Robert) Homme(David) relParentEnfant(Sophie,Anne) relParentEnfant(Robert,David)

Une base de connaissances composée d'une Tbox et d'une Abox

Le concept $\top$ et le rôle $\top_R$ sont les plus généraux de leur catégorie, le concept $\perp$ ainsi que $\perp_R$ les plus spécifiques.

Les concept atomiques et rôles atomiques constituent les entités élem d'une Tbox.. (concept en magiscule, rôle en miniscule)

Subsomption

Les deux inférences principales dans les DL sont la **classification** de concepts qui s'effectue au niveau de la T-Box, et la **reconnaissance d'instances** qui s'effectue au niveau de la composante assertionnelle.

Ces deux opérations sont basées sur des calculs de relations de subsomption.

Subsomption

Relation qui permet d'ordonner les concepts d'une DL dans une hiérarchie.

Le concept être-Humain subsume le concept Enfant

Différents type de subsomption

- **Extensionnelle** : un concept C subsume un concept D Ssi l'ensemble des instances de C contient l'ensemble des instances de D (utilisée dans la théorie pour la détermination de la correction et de la complétude). Pour les calculs effectifs, d'autres types de subsumption sont utilisés
- **Structurelle (ou intensionnelle**) : un concept C subsume un concept D Ssi l'ensemble de ses propriétés est inclus dans l'ensemble des propriétés de D. Par exemple le concept Enfant contient l'ensemble des propriétés du concept être-Humain. Le concept Enfant possède toutes les propriétés du concept être humain plus des propriétés qui lui sont spécifiques (ex : age <12). Le calcul se fait à partir de structures qui sont des représentations abstraites le plus souvent canoniques des propriétés du concept.

Inférences terminologiques

Classification de concepts : placer un concept défini à la place la plus appropriée dans la hiérarchie. Ce processus permet de découvrir les relations de subsomption qui existent entre un nouveau concept et les concepts déjà dans la taxonomie (les *subsumeurs* puis les *subsumés*).

Héritage : Le mécanisme d'héritage consiste à retrouver toutes les propriétés d'un concept à partir des propriétés des concepts qui le subsument, et la **Complétion** permet de retrouver les propriétés déduites logiquement.

Inférences assertionnelles

Opération de reconnaissance d'instances (opération principale) : trouver pour un individu donné les concepts les plus spécifiques dont il est instance. L'opération élémentaire utilisée dans cette recherche est l'opération de *test d'instance (instance checking)* : *o* est une instance du concept *C*

Une des méthodes employées pour faire de la *reconnaissance d'instances* est de calculer l'*abstraction* d'un objet et de classer le concept obtenu.

Pour calculer le *concept abstrait* d'un individu, on retrouve toutes les informations liées à l'individu et on les rassemble sous la forme d'une définition de concept la plus spécifique possible.

Pour savoir si un objet o_1 est une instance d'un concept *C*, on calcule le *concept abstrait* A_{01}, on teste ensuite si *C* subsume A_{01}. Si c'est le cas, on déduit alors que o_1 est une instance du concept *C*.

Autres opérations assertionnelles

Fermeture de rôles : si r est fermé pour un individu I, on sait qu'il n'existe pas d'autres *filler* (il n'y a pas d'autres individus en relation par r avec I) que ceux exprimés explicitement. Cette opération peut modifier le calcul du concept abstrait d'un individu et donc le résultat de l'opération de reconnaissance d'instances.

Ex : soit le concept chien-abandonné (chien qui n'ont pas de maître) défini comme suit :

Chien-abandonné ≡ *chien and atmost 0 maître*.

Atmost n r est un connecteur qui restreint la cardinalité maximale d'un rôle *r* à la valeur *n*, n étant un entier.

Soit l'unique assertion concernant *o1*:

Chien(o1).

On exprime ici le fait que *o1* est un chien.

Si le rôle *maître* n'est pas fermé pour *o1*, le concept abstrait de *o1* est le suivant :

A_{01} = *chien.* A la questi*on 'o1* est-il une instance du concept *chien-abandonné?'*, la réponse est *non* car *atmost 0 maître* n'est pas une propriété de A_{01} .

Si on suppose maitenant que le rôle *maître* est fermé pour *o1*, le concept abstrait de *o1* devient *:* A_{01} = *chien and atmost 0 maître.*

En effet, on sait que *o1* est en relation par maître avec aucun individu de la A-Box. On reconnaîtra alors *o1* comme une instance du concept *chien-abandonné.*

Propagation:

Une assertion sur un individu o1 peut avoir des conséquences logique sur des individus en relation avec *o1*. Les DL qui bénéficient de l'opération de *propagation* propagent les nouvelles informations déduites sur les individus concernés.

Détection de contradiction :

Il y a contradiction dans une A-Box lorsque des faits concernant un individu sont contradictoires. Certaine A-Box disposent d'une opération appelée détection de contradiction.

Chaînage avant :

Certaines A-Box permettent de définir des règles ayant des concepts en prémisse et en conclusion. Dans ce cas, si on a trouvé qu'un individu satisfait un concept se trouvant en prémisse d'une règle, on en déduit qu'il satisfait aussi le concept se trouvant en conclusion de la règle.

Complexité

Certaines DL favorisent la puissance d'expression au détriment de la complétude de leurs algorithmes d'inférence.

Ceci posent un certain pb de complexité.

Pour un langage de base (and, not concept atomique, all, some, $\top$, $\bot$),la complexité est polynomiale.

La disjonction de concepts et la conjonction de rôles sont sources de complexité. Leur introduction dans des langages rend la complexité NP-difficile.

Il ne faut pas forcément rejeter les langages dont la complexité n'est pas polynomiale. En pratique, ils restent utilisables, tout dépend des buts que se fixent les concepteurs d'une DL

Domaines d'application

1. Web-based information systems. Malgré le succès du www, il reste néanmoins limité à cause de ses liens avec des langages tel que HTML qui se focalisent sur la présentation (formater des textes) plutôt que sur le contenu.

Des langages tels que XML ont tenté de capturer un peu plus le sens mais ils sont très insuffisants et ne peuvent supporter des applications intelligentes. Or il est urgent que ces applications se développent. En effet, même si nous sommes très contents de taper quelques mots clés sur Google pour obtenir des informations sur un sujet qui nous intéressent , très souvent nous sommes frustrés par le manque d'informations (**le silence**) ou agacés par le flot d'information inutiles (**les bruits**). L'objectif est d'introduire plus de sémantique pour capturer les informations du web. Pour cela, des langages basés sur des DL ont été développés, OWL (OWL DL) par exemple (web ontology language 2004) est le world wide web consortium (W3C) langage recommandé pour le web semantic. Il exploite la force des DL en particulier sa sémantique et ses techniques de raisonnement.

2. Ingénierie des logiciels : l'idée est d'utiliser une DL pour implémenter un système qui supporte le développeur de logiciels en l'aidant à retrouver des informations sur un gros système de logiciel (LaSSIE system Devanbu et al. 1991)
3. Langage Naturel. Le traitement du langage naturel est l'application à l'origine de cette logique : le sens des mot ou lexique, représentation du sens d'une phrase, le contexte, la désambigüisation des différentes lectures de phrases etc…
4. Gestion des bases de données : le liens avec les BD est très fort. On a souvent besoin de faire coexister des SGBD et des KBS. Le premier pour la persistance des données et la gestion d'un grand nombre de ses données le second pour la gestion intentionnelle des connaissances.

Défauts et exceptions dans les DL

1. [J. Quantz &V. Royer 92] : étendre les DL en y incluant une forme limitée de raisonnement par défaut. Les défauts ne sont pas autorisés dans la définition même des concepts mais en tant que règles incidentes des concepts. Le but principal est la spécification d'une sémantique basée sur les sémantiques de préférence définies dans [Shoham 88]. Les défauts sont exprimés sous la forme de règles du type : c1~ c2. Cette règle signifie : si un objet est une instance du concept C1 alors c'est aussi une instance du concept C2 à moins d'un conflit avec une autre connaissance. (un conflit est caractérisé par l'appartenance à deux concept disjoints, i.e., intersection de leurs extensions vide). Si le conflit porte sur des propriétés strictes, l'information posant pb est rejetée. Si un objet est une instance par défaut de deux concepts disjoints, les auteurs supposent qu'il y a exception sur un des défauts et lancent un processus de résolution de conflits. Ce processus est basé sur la théorie des modèles préférentiels de Shoham.
2. [L. Padgham & al.93] s'inspirent des travaux réalisés dans le cadre de l'*héritage non monotone*. Ils définissent la notion de *subsomption défaut* en se basant sur *l'héritage sceptique*. Dans leur approche un concept possède une définition scindée en deux parties: une partie (appelée *core*) qui comprend les propriétés strictes suffisantes et nécessaires du concept et une partie (appelée *default*) qui comprend des propriétés strictes mais aussi des propriétés défaut.

3. P. Coupey et C. Fouqueré qui ont permis l'introduction de deux nouveaux connecteurs (δ, ε) au sein même de la définition des concepts (dans la T-Box). De manière intuitive, le connecteur δ représente la notion de *défaut* comme par exemple la définition suivante : *Mammifère* ≡ *Animal* ∩ δ *Vivipare* ∩ *Vertébré* définit le concept *Mammifère* comme un Animal Vertébré généralement Vivipare.

Malheureusement, en utilisant cette définition de Mammifère on peut par exemple inférer qu'un *Canard* (*Canard* ≡ *Animal* ∩ *Ovipare* ∩ *Vertébré* ∩ *Avec-bec* ∩ *Pied-Palmé* ∩ *Vol)* est un mammifère puisqu'il possède les propriétés Vertébré et Animal. En présence de connaissances par défaut la classification automatique semble impossible (def non nécessaire).

Pour résoudre ce pb, les auteurs introduisent le connecteur ε qui représente une exception à un concept. Ils définissent un point de vue définitionnel pour les défauts et expriment la propriété suivante :

Un objet est une instance d'un concept C ssi il satisfait les propriétés strictes de C ou est explicitement ''exceptionnel '' par rapport aux propriétés défauts de C.

Avec cette propriété on ne peut plus inférer qu'un canard est un Mammifère puisqu'il n'est ni vivipare ni exceptionnel par rapport au fait d'être vivipare par contre le concept *Ornithorynque* (*Ornithorynque* ≡ *Animal* ∩ *Vertébré* ∩ *Ovipare* ∩ *Avec-bec* ∩ *Vivipare*$^{\varepsilon}$) sera classé sous le concept *Mammifère* vérifie les propriété strictes de *Mammifère* et est exceptionnel / au fait d'être *Vivipare.*