

Logique de Description

Description Logics

Guy Lapalme
Université de Montréal

Approches pour la représentation de connaissances

- logique
 - logique des prédicats
 - raisonnement: vérifier la conséquence logique
- non logique (inspiration cognitive)
 - réseaux sémantiques
 - règles tirées d'expériences psychologiques
 - raisonnement: procédure ad-hoc de manipulation de structures

Logique de description

- combine les approches logiques ou non logiques
- a débuté sous le nom de *Terminological Systems*
 - emphase sur l'établissement de la terminologie du domaine représenté
- a aussi comme nom:
 - Concept Languages
 - Term Subsumption Languages
 - KL-ONE based Languages

Logique de description

- définit les concepts pertinents d'un domaine (terminologie)
- utilise les concepts pour spécifier les propriétés des objets, des individus dans le domaine
- a une sémantique logique formelle pour déduire des connaissances implicites via la classification de concepts et d'individus

Composantes DL

- **individus ou objets (constantes)**
identifiables par un nom
exemples: MARY, PETER, PAUL
- **concepts atomiques (prédicats unaires)**
classes d'objets ou ensemble de propriétés
exemples: Woman, Human, Mother
- **rôles atomiques (prédicats binaires)**
relations entre les objets
exemples: hasChild, hasHusband
- **constructeurs**
exemples: \neg , \sqcap , \sqcup , \sqsubseteq , \exists , \forall

Mécanismes d'inférence

- Soit les définitions suivantes

- Woman \sqsubseteq Person	TBox
- Mother \equiv Woman \sqcap \exists hasChild.Person	

- hasChild(MARY, PETER)	ABox
- Woman(MARY)	
- Person(PETER)	

- Déduire de nouveaux faits
 - Person(MARY)
 - Mother(MARY)

Inférences DL - I

TBox: axiomes sur les concepts

- Un concept est-il consistant ?
 - peut-il exister au moins une instance d'une classe ?
- *Subsommption*: un concept peut-il être déduit?
 - $\text{Woman} \sqsubseteq \text{Person} \Rightarrow \text{Woman} \sqsubseteq \text{Animal}$
 - $\text{Person} \sqsubseteq \text{Animal}$

Inférences DL - 2

ABox: assertions sur les individus

- vérification de la consistance d'un ensemble d'assertions
 - un individu déclaré comme instance d'une classe répond-il à ses caractéristiques ?
p.e. une mère sans enfant
 - une relation déclarée entre deux individus est-elle possible ?
p.e. célibataire marié à un autre individu

Un contenu de TBox

Figure 2.2 de F. Baader, W. Nutt

$Woman \equiv Person \sqcap Female$

$Man \equiv Person \sqcap \neg Woman$

$Mother \equiv Woman \sqcap \exists hasChild. Person$

$Father \equiv Man \sqcap \exists hasChild. Person$

$Parent \equiv Father \sqcup Mother$

$Grandmother \equiv Mother \sqcap \exists hasChild. Parent$

$MotherWithManyChildren \equiv Mother \sqcap \geq 3 hasChild$

$MotherWithoutDaughter \equiv Mother \sqcap \forall hasChild. \neg Woman$

$Wife \equiv Woman \sqcap \exists hasHusband. Man$

Un contenu de ABox

Figure 2.4 de F. Baader, W. Nutt

MotherWithoutDaughter (MARY)

Father (PETER)

hasChild (MARY, PETER)

hasChild (MARY, PAUL)

hasChild (PETER, HARRY)

Combinaison

Woman \equiv Person \sqcap Female

TBox

Man \equiv Person \sqcap \neg Woman

Mother \equiv Woman \sqcap \exists hasChild.Person

Father \equiv Man \sqcap \exists hasChild.Person

Parent \equiv Father \sqcup Mother

Grandmother \equiv Mother \sqcap \exists hasChild.Parent

MotherWithManyChildren \equiv Mother \sqcap ≥ 3 hasChild

MotherWithoutDaughter \equiv Mother \sqcap \forall hasChild. \neg Woman

Wife \equiv Woman \sqcap \exists hasHusband.Man

MotherWithoutDaughter(MARY)

ABox

Father(PETER)

hasChild(MARY, PETER)

hasChild(MARY, PAUL)

hasChild(PETER, HARRY)

\Rightarrow Grandmother(MARY)

Exemple de langage de description (\mathcal{AL})

Attributive Language

- langage minimal mais pratique
- concept atomique : A
- rôle atomique : R
- description de concept :
 $C, D ::= A \mid \perp \mid \top \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R.T$
 - négation que de concepts atomiques
 - seul \top est utilisable dans l'existence
- traduction possible en logique des prédicats

Expressions \mathcal{AL}

Person *concept atomique*

Female *concept atomique*

Person \sqcap Female *une femme*

Person \sqcap \neg Female *un homme?*

hasChild *rôle*

Person \sqcap \exists hasChild. \top *personnes avec au moins un enfant*

Person \sqcap \forall hasChild. \perp *personnes sans enfants*

Person \sqcap \forall hasChild.Female *personnes qui n'ont que des filles*

Interprétation par des ensembles

$\Delta^I =$ domaine d'interprétation

$$A^I \subseteq \Delta^I, R^I \subseteq \Delta^I \times \Delta^I$$

$$\top^I = \Delta^I$$

$$\perp^I = \emptyset^I$$

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

$$(C \sqcap D)^I = C^I \cap D^I$$

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

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

$$(C \equiv D)^I = C^I = D^I$$

Exemple d'interprétation

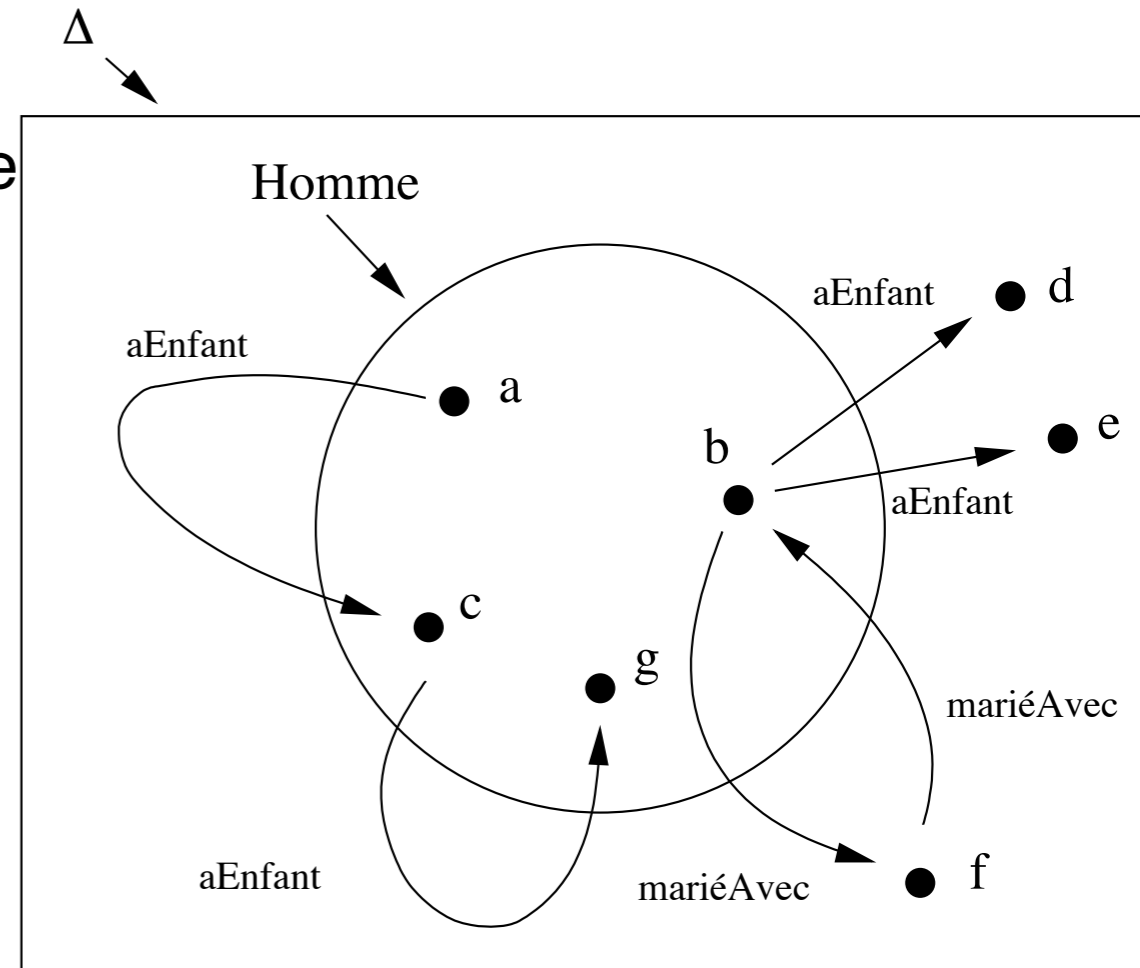
Parent $\equiv \exists a\text{Enfant}.\top$

ParentDeFemme $\equiv \exists a\text{Enfant}.\top \sqcap \forall a\text{Enfant}.\neg\text{Homme}$

Célibataire $\equiv \forall\text{mariéAvec}.\perp$

HommeMarié $\equiv \text{Homme} \sqcap \exists\text{mariéAvec}.\top$

GrandParentChoyé $\equiv \forall a\text{Enfant}.\left(\exists a\text{Enfant}.\top\right)$



Parent : $\{a, b, c\}$

ParentDeFemme : $\{b\}$

Célibataire : $\{a, c, d, e, g\}$

HommeMarié : $\{b\}$

GrandParentChoyé : $\{a, d, e, f, g\}$

$\Delta = \{a, b, c, d, e, f, g\}$

$\mathcal{I}(\text{Homme}) = \{a, b, c, g\}$

$\mathcal{I}(\text{aEnfant}) = \{(a, c), (b, d), (b, e), (c, g)\}$

$\mathcal{I}(\text{mariéAvec}) = \{(b, f), (f, b)\}$

Autre interprétation possible!

Parent $\equiv \exists a \text{Enfant}.\top$

ParentDeFemme $\equiv \exists a \text{Enfant}.\top \sqcap \forall a \text{Enfant}.\neg \text{Homme}$

Célibataire $\equiv \forall \text{mariéAvec}.\perp$

HommeMarié $\equiv \text{Homme} \sqcap \exists \text{mariéAvec}.\top$

GrandParentChoyé $\equiv \forall a \text{Enfant} . (\exists a \text{Enfant} . \top)$

$\Delta = \{a\}$

$\mathcal{I}(\text{Homme}) = \{\}$

$\mathcal{I}(a\text{Enfant}) = \{\}$

$\mathcal{I}(\text{mariéAvec}) = \{\}$

Familles \mathcal{AL} - I

C : négations de concepts arbitraires

$$(\neg C)^I = \Delta^I \setminus C^I$$

\sqcup : Disjonction

$$(C \sqcup D)^I = C^I \cup D^I$$

E : quantification existentielle complète

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

F : fonction plutôt que relation (p.e. mariéAvec)

I : inversion (p.e. enfantDe \equiv parentDe⁻)

Familles \mathcal{AL} - 2

\mathcal{N} : Restriction de cardinalité

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

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

p.e. `Mother \sqcap ≥ 3 hasChild`

\mathcal{Q} : Restriction de cardinalité qualifiée

$$(\cong_n \mathbf{R.C})^I = \{a \in \Delta^I \mid |\{b \mid (a,b) \in R^I \wedge b \in C^I\}| \geq n\}$$

$$(\leq_n \mathbf{R.C})^I = \{a \in \Delta^I \mid |\{b \mid (a,b) \in R^I \wedge b \in C^I\}| \leq n\}$$

p.e. `Mother \sqcap ≥ 3 hasChild.Woman`

Familles \mathcal{AL} - 3

O : Nominaux : énumération d'individus

$$(\{a_1, \dots, a_n\})^I = \{(a_1)^i, \dots, (a_n)^i\}$$

\mathcal{R} : Inclusion de rôles : individus reliés à un autre

$$(R:a)^I = \{d \in \Delta \mid (d, a^i) \in R^I\}$$

p.e. `hasChild:PETER`

Familles de logique

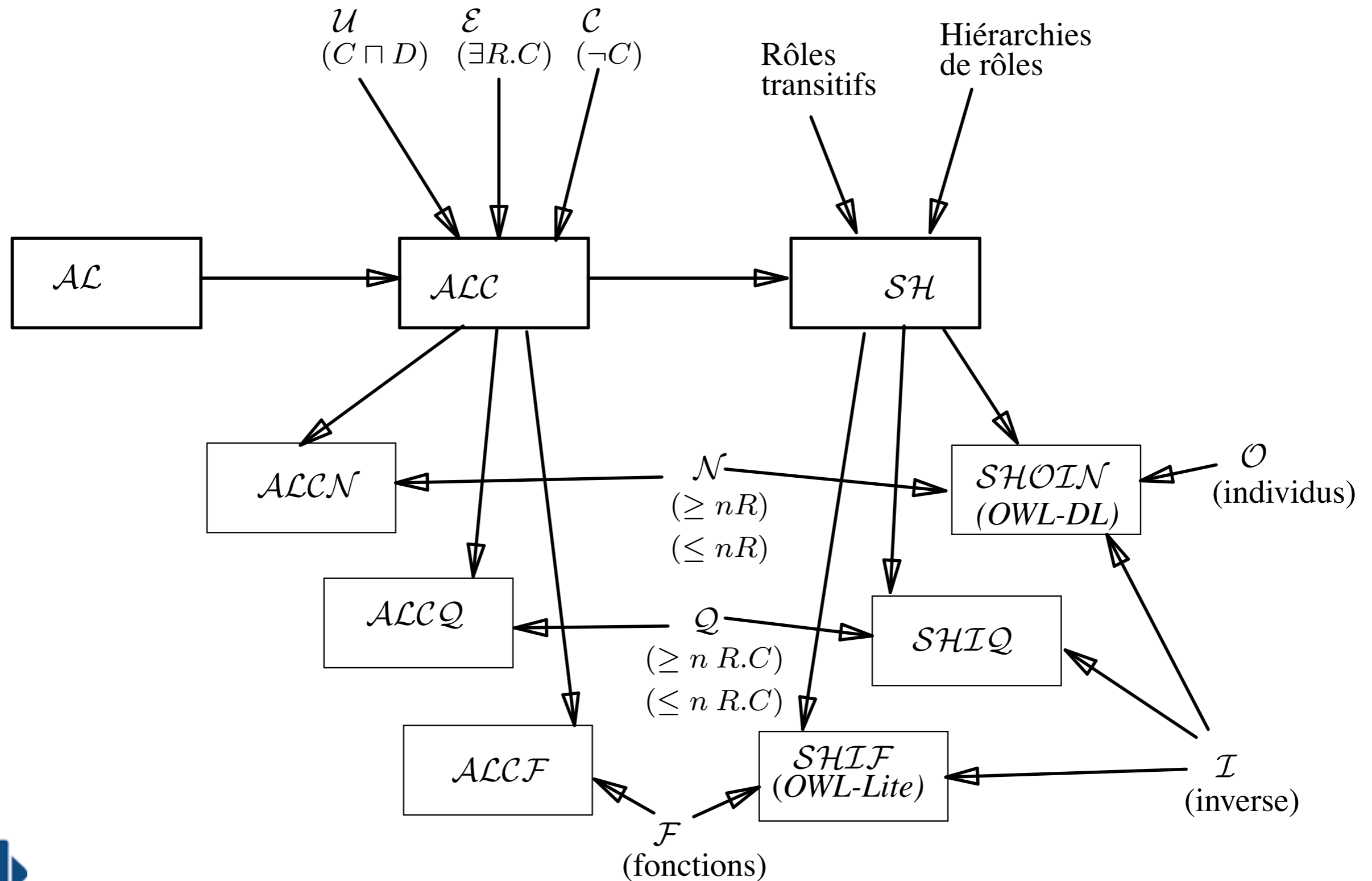


Figure tirée de : Michel Gagnon, *Logique descriptive et OWL*, École polytechnique, Montréal, 2012.

Constructions OWL en \mathcal{ALC}

$C, D ::= A \mid \perp \mid \top \mid \neg C \mid C \sqcap D \mid C \sqcup D \mid \forall R.C \mid \exists R.C$

- classes, rôles, individus
- appartenance à des classes
- instances de rôles
- `owl:Thing` et `owl:Nothing`
- inclusion, disjonction et équivalence de classes
- restrictions de rôles: `owl:allValuesFrom`,
`owl:someValuesFrom`
- `rdfs:domain`, `rdfs:range`

Noms des extensions de \mathcal{AL}

\mathcal{U}	Union
\mathcal{C}	complément
\mathcal{E}	restrictions existentielles de rôle
\mathcal{F}	rôles fonctionnels
\mathcal{I}	rôles inverses
\mathcal{N}	restrictions de cardinalité
\mathcal{Q}	restrictions qualifiées de cardinalité
\mathcal{O}	nominaux
\mathcal{R}	axiomes d'inclusion de rôle
\mathcal{S}	\mathcal{ALC} + transitivité des rôles
\mathcal{H}	hiérarchie des rôles
\mathcal{D}	types de données

OWL vs DL

Full	\emptyset
DL	$\mathcal{SHOIN}(\mathcal{D})$
DL-Lite	$\mathcal{SHIF}(\mathcal{D})$
2 Full	\emptyset
2 DL	$\mathcal{SROIQ}(\mathcal{D})$
2 EL	\mathcal{EL}
2 QL	DL-Lite
2 RL	DLP

Navigateur de complexité de DL

Expression du nouvel \mathcal{AL}

Person \sqcap (\leq_1 hasChild \sqcup
(\geq_3 hasChild \sqcap \exists hasChild.Female))

les personnes avec au plus un enfant ou avec 3 enfants et au moins une fille

DL peut aussi être traduit en logique des prédicats - I

Fig 5.7 de Hitzler et al.,2010

concept \equiv prédicat unaire avec variable libre

$$\pi_x(A) = A(x)$$

rôle \equiv prédicat primitif binaire

$$\pi_{x,y}(R) = R(x,y)$$

inclusion de classes

$$\pi_x(C \sqsubseteq D) = \forall x (\pi_x(C) \rightarrow \pi_x(D))$$

\sqcap , \sqcup , \neg \equiv interprétation ensembliste habituelle

$$\pi_x(C \sqcap D) = \pi_x(C) \wedge \pi_x(D)$$

$$\pi_x(C \sqcup D) = \pi_x(C) \vee \pi_x(D)$$

$$\pi_x(\neg C) = \neg \pi_x(C)$$

DL peut aussi être traduit en logique des prédicats - 2

$$\pi_x(\forall R.C) = \forall x'(R(x,x') \rightarrow \pi_{x'}(C))$$

$$\pi_x(\exists R.C) = \exists x'(R(x,x') \wedge \pi_{x'}(C))$$

$$\pi_x(\geq n S.C) =$$

$$\exists x_1 \dots x_n (\bigwedge_{i \neq j} (x_i \neq x_j) \wedge \bigwedge_i (s(x, x_i) \wedge \pi_{x_i}(C)))$$

$$\pi_x(\leq n S.C) =$$

$$\neg \exists x_1 \dots x_{n+1} (\bigwedge_{i \neq j} (x_i \neq x_j) \wedge \bigwedge_i (s(x, x_i) \wedge \pi_{x_i}(C)))$$

$$\pi_x(\{a\}) = (x=a)$$

DL peut aussi être traduit en logique des prédicats - 3

$$\pi(R_1 \sqsubseteq R_2) = \forall x \forall y (\pi_{x,y}(R_1) \rightarrow \pi_{x,y}(R_2))$$

$$\pi_{x,y}(S) = S(x,y)$$

$$\pi_{x,y}(R^-) = \pi_{y,x}(R)$$

$$\pi_{x,y}(R_1 \bullet \dots \bullet R_n) = \exists x_1 \dots x_{n-1}$$

$$(\pi_{x,x_1}(R_1) \wedge \bigwedge_{i=1..n-2} \pi_{x_i,x_{i+1}}(R_{i+1}) \wedge \pi_{x_{n-1},y}(R_n))$$

$$\pi(\text{Ref}(R)) = \forall x \pi_{x,x}(R)$$

$$\pi(\text{Asy}(R)) = \forall x \forall y (\pi_{x,y}(R) \rightarrow \neg \pi_{y,x}(R))$$

$$\pi(\text{Dis}(R_1, R_2)) = \neg \exists x \exists y (\pi_{x,y}(R_1) \wedge \pi_{x,y}(R_2))$$

Inférences possibles avec la logique des prédicats, mais il y a des algorithmes plus efficaces de résolution sans variable

Exemples de traductions DL en logique des prédicats - I

Fig 5.9 de Hitzler et al.,2010

- Professor \sqsubseteq FacultyMember
 $\forall x(\text{Professor}(x) \rightarrow \text{FacultyMember}(x))$
- Professor \sqsubseteq (Person \sqcap FacultyMember)
 \sqcup (Person \sqcap \neg PhDStudent)
 $\forall x(\text{Professor}(x) \rightarrow (\text{Person}(x) \wedge \text{FacultyMember}(x))$
 $\vee (\text{Person}(x) \wedge \neg \text{PhDStudent}(x)))$
- Exam \sqsubseteq \forall hasExaminer.Professor
 $\forall x (\text{Exam}(x) \rightarrow \forall y(\text{hasExaminer}(x,y) \rightarrow \text{Professor}(y)))$

Exemples de traductions DL en logique des prédicats - 2

- $\text{Exam} \sqsubseteq \leq 2 \text{hasExaminer}$

$$\forall x (\text{Exam}(x) \rightarrow \neg \exists x_1, x_2, x_3 (x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_1 \neq x_3) \\ \wedge \text{hasExaminer}(x, x_1) \wedge \text{hasExaminer}(x, x_2) \\ \wedge \text{hasExaminer}(x, x_3))$$

- $\text{hasParent} \cdot \text{hasBrother} \sqsubseteq \text{hasUncle}$

$$\forall x \forall y (\exists x' \text{hasParent}(x, x') \wedge \text{hasBrother}(x', y) \\ \rightarrow \text{hasUncle}(x, y))$$

Inférences avec TBox

- Satisfiabilité : existe-il un modèle pour C ?
- Subsumption entre classes
 - est-ce que $C \sqsubseteq D$? (C est subsumé par D)
 - mère est toujours un parent
- Équivalence de classes ($C \equiv D$)
- Classes disjointes ($C \sqcap D \sqsubseteq \perp$)

Tous réduisibles à insatisfiabilité, par

- $C \sqsubseteq D \Leftrightarrow C \sqcap \neg D$ insatisfaisable
- $C \equiv D \Leftrightarrow C \sqcap \neg D$ et $\neg C \sqcap D$ insatisfaisables
- $C \sqcap D \sqsubseteq \perp \Leftrightarrow C \sqcap D$ insatisfaisable

Inférences avec ABox

- Consistance de description d'individus
 - aucune instance $C(a)$ s'il est impossible que a appartienne à C
- Vérification d'instances
 - est-ce qu'un individu a appartient à C ? $C(a)$
- Trouver toutes les instances d'une classe C
- Trouver des classes correspondant à des individus
 - Marie est une femme avec un enfant, alors elle est mère
- Détermination de la taxonomie de classes

Preuve par tableau

- Logique des propositions
 - voir document de J. Bergeron
- Logique de description
 - éliminer la TBox en remplaçant les termes par leur définition
 - transformer en *Negation Normal Form* (NNF)
 - enlever les \sqsubseteq ($C \sqsubseteq D = \neg C \sqcup D$)
 - négation en face des noms de classes
 - montrer que la négation de ce qu'on veut prouver est insatisfaisable

Transformation en NNF (\mathcal{N})

Fig 5.10 de Hitzler et al.,2010

$$\mathcal{N}(C \sqsubseteq D) \Rightarrow \mathcal{N}(\neg C \sqcup D)$$

$$\mathcal{N}(C) \Rightarrow C$$

$$\mathcal{N}(\neg C) \Rightarrow \neg C$$

$$\mathcal{N}(\neg\neg C) \Rightarrow \mathcal{N}(C)$$

$$\mathcal{N}(C \sqcup D) \Rightarrow \mathcal{N}(C) \sqcup \mathcal{N}(D)$$

$$\mathcal{N}(C \sqcap D) \Rightarrow \mathcal{N}(C) \sqcap \mathcal{N}(D)$$

$$\mathcal{N}(\neg(C \sqcup D)) \Rightarrow \mathcal{N}(\neg C) \sqcap \mathcal{N}(\neg D)$$

$$\mathcal{N}(\neg(C \sqcap D)) \Rightarrow \mathcal{N}(\neg C) \sqcup \mathcal{N}(\neg D)$$

$$\mathcal{N}(\forall R.C) \Rightarrow \forall R.\mathcal{N}(C)$$

$$\mathcal{N}(\exists R.C) \Rightarrow \exists R.\mathcal{N}(C)$$

$$\mathcal{N}(\neg\forall R.C) \Rightarrow \exists R.\mathcal{N}(\neg C)$$

$$\mathcal{N}(\neg\exists R.C) \Rightarrow \forall R.\mathcal{N}(\neg C)$$

$$\mathcal{N}(\leq n R.C) \Rightarrow \leq n R.\mathcal{N}(C)$$

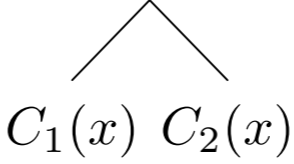
$$\mathcal{N}(\geq n R.C) \Rightarrow \geq n R.\mathcal{N}(C)$$

$$\mathcal{N}(\neg\leq n R.C) \Rightarrow \geq (n+1)R.\mathcal{N}(C)$$

$$\mathcal{N}(\neg\geq (n+1)R.C) \Rightarrow \leq n R.\mathcal{N}(C)$$

$$\mathcal{N}(\neg\geq 0R.C) \Rightarrow \perp$$

Règles de tableau pour DL

Règle	Condition	Action
règle- \sqcap	\mathcal{A} contient $(C_1 \sqcap C_2)(x)$ et ne contient pas déjà les deux énoncés $C_1(x)$ et $C_2(x)$.	On ajoute à \mathcal{A} les énoncés $C_1(x)$ et $C_2(x)$
règle- \sqcup	\mathcal{A} contient $(C_1 \sqcup C_2)(x)$ et ne contient aucun des deux énoncés $C_1(x)$ et $C_2(x)$.	On ajoute à \mathcal{A} le branchement suivant : 
règle- \exists	\mathcal{A} contient $(\exists R.C)(x)$ et il n'existe aucun individu z tel que $R(x, z)$ et $C(z)$ sont aussi dans \mathcal{A}	On ajoute $R(x, y)$ et $C(y)$ à \mathcal{A} , où y est un nom d'individu qui n'existe pas déjà dans \mathcal{A}
règle- \forall	\mathcal{A} contient $(\forall R.C)(x)$ et $R(x, y)$, mais ne contient pas $C(y)$	On ajoute $C(y)$ à \mathcal{A}
règle- \geq	\mathcal{A} contient $(\geq n R)(x)$ et il n'y a pas dans \mathcal{A} des individus z_1, \dots, z_n qui sont tous distincts (c'est-à-dire qu'on doit avoir explicitement dans \mathcal{A} l'énoncé $z_i \neq z_j$ pour chaque paire possible avec cet ensemble d'individus) et qui sont tels que \mathcal{A} contient la relation $R(x, z_i)$ pour tous ces individus ($1 \leq i \leq n$).	Soit un ensemble de n individus dénotés par y_1, \dots, y_n , qui sont des noms qui n'existent pas dans \mathcal{A} . On ajoute à \mathcal{A} les énoncés $y_i \neq y_j$ pour chaque paire possible avec cet ensemble, ainsi que les énoncés $R(x, y_i)$ pour ($1 \leq i \leq n$).
règle- \leq	\mathcal{A} contient $(\leq n R)(x)$ et les énoncés $R(x, y_1), \dots, R(x, y_{n+1})$. Il n'existe aucune identité $y_i = y_j$ dans \mathcal{A} pour ($1 \leq i \leq n + 1$), ($1 \leq j \leq n + 1$), $i \neq j$	Pour chaque paire possible (y_i, y_j) d'individus parmi y_i, y_{n+1} , on ajoute une nouvelle branche avec $y_i \neq y_j$.

Exemple de preuve par tableau

Woman \equiv Person \sqcap Female

TBox

Man \equiv Person \sqcap \neg Woman

Mother \equiv Woman \sqcap \exists hasChild.Person

Father \equiv Man \sqcap \exists hasChild.Person

Parent \equiv Father \sqcup Mother

Grandmother \equiv Mother \sqcap \exists hasChild.Parent

MotherWithManyChildren \equiv Mother \sqcap ≥ 3 hasChild

MotherWithoutDaughter \equiv Mother \sqcap \forall hasChild. \neg Woman

Wife \equiv Woman \sqcap \exists hasHusband.Man

MotherWithoutDaughter(MARY)

ABox

Father(PETER)

hasChild(MARY, PETER)

hasChild(MARY, PAUL)

hasChild(PETER, HARRY)

Person(PAUL). Person(HARRY)

\Rightarrow Grandmother(MARY) ?

Remplacement des définitions de la TBox

en absence de définitions circulaires

Adaptation de Figure 2.3 de F. Baader, W. Nutt

Woman \equiv Person \sqcap Female

Man \equiv Person \sqcap \neg (Person \sqcap Female)

Mother \equiv (Person \sqcap Female) \sqcap \exists hasChild.Person

Father \equiv (Person \sqcap \neg (Person \sqcap Female)) \sqcap \exists hasChild.Person

Parent \equiv ((Person \sqcap \neg (Person \sqcap Female)) \sqcap \exists hasChild.Person) \sqcup ((Person \sqcap Female) \sqcap \exists hasChild.Person)

Grandmother \equiv ((Person \sqcap Female) \sqcap \exists hasChild.Person) \sqcap \exists hasChild.(((Person \sqcap \neg (Person \sqcap Female)) \sqcap \exists hasChild.Person) \sqcup ((Person \sqcap Female) \sqcap \exists hasChild.Person))

MotherWithManyChildren \equiv ((Person \sqcap Female) \sqcap \exists hasChild.Person) \sqcap ≥ 3 hasChild

MotherWithoutDaughter \equiv Person \sqcap Female \sqcap \forall hasChild. (\neg Person \sqcup \neg Female)

Transformation en NNF

Woman \equiv Person \sqcap Female

Man \equiv Person \sqcap \neg Female

Mother \equiv Person \sqcap Female \sqcap \exists hasChild.Person

Father \equiv Person \sqcap \neg Female \sqcap \exists hasChild.Person

Parent \equiv Person \sqcap \exists hasChild.Person

Grandmother \equiv Person \sqcap Female \sqcap \exists hasChild.Person \sqcap
 \exists hasChild.(Person \sqcap \exists hasChild.Person)

MotherWithMany
Children \equiv Person \sqcap Female \sqcap ≥ 3 hasChild

MotherWithout
Daughter \equiv Person \sqcap Female \sqcap
 \forall hasChild.(\neg Person \sqcup \neg Female)

Opérations sur la ABox

$\text{Person}(\text{MARY})$ ← Fermer branche 1
 $\text{Female}(\text{MARY})$ ← Fermer branche 2
 $\exists \text{hasChild}.\text{Person}(\text{MARY})$
 $\text{Person}(\text{PETER})$
 $\neg \text{Female}(\text{PETER})$
 $\text{hasChild}(\text{PETER}, w) \cdot \text{Person}(w)$
 $\text{hasChild}(\text{MARY}, \text{PETER})$
 $\text{hasChild}(\text{MARY}, \text{PAUL})$
 $\text{hasChild}(\text{PETER}, \text{HARRY})$

règle- \exists \mathcal{A} contient $(\exists R.C)(x)$ et il n'existe aucun individu z tel que $R(x, z)$ et $C(z)$ sont aussi dans \mathcal{A}
 On ajoute $R(x, y)$ et $C(y)$ à \mathcal{A} , où y est un nom d'individu qui n'existe pas déjà dans \mathcal{A}

Ajout de la négation de ce qu'on veut prouver

$\neg \text{Grandmother}(\text{MARY})$
 $\neg \text{Person}(\text{MARY}) \sqcup \neg \text{Female}(\text{MARY}) \sqcup$
 $\neg \exists \text{hasChild}.\text{Person}(\text{MARY}) \sqcup$
 $\neg (\exists \text{hasChild}.\text{Person} \sqcap \exists \text{hasChild}.\text{Person})(\text{MARY})$

Fermer la troisième branche

$\neg \exists \text{hasChild. Person (MARY)}$

$(\forall \text{hasChild. } \neg \text{Person}) (\text{MARY})$

$\neg \text{Person (PAUL)}$

$\neg \text{Person (PETER)}$

règle- \forall \mathcal{A} contient $(\forall R.C)(x)$ et $R(x,y)$, mais ne
contient pas $C(y)$

On ajoute $C(y)$ à \mathcal{A}

car ABox contient

$\text{hasChild (MARY, PAUL)}$

$\text{hasChild (MARY, PETER)}$

Ferme car

Person (PAUL) dans ABox

Person (PETER) dans TBox transformé

Fermer la quatrième branche

$\neg(\exists \text{hasChild} . (\text{Person} \sqcap \exists \text{hasChild} . \text{Person})) (\text{MARY})$
 $(\forall \text{hasChild} . (\neg(\text{Person} \sqcap \exists \text{hasChild} . \text{Person}))) (\text{MARY})$
 $(\forall \text{hasChild} . (\neg \text{Person} \sqcup \neg \exists \text{hasChild} . \text{Person})) (\text{MARY})$
 $(\forall \text{hasChild} . (\neg \text{Person} \sqcup \forall \text{hasChild} . \neg \text{Person})) (\text{MARY})$
 $(\forall \text{hasChild} . \neg \text{Person}) (\text{MARY}) \sqcup$
 $((\forall \text{hasChild} . (\forall \text{hasChild} . \neg \text{Person})) (\text{MARY}))$

$\neg \text{Person} (\text{PAUL})$

$\neg \text{Person} (\text{PETER})$

Règle \forall (idem que 3e branche)

ferme cette sous-branche

Fermer la 2e sous-branche

$((\forall \text{hasChild} . (\forall \text{hasChild} . \neg \text{Person})) (\text{MARY}))$

$((\forall \text{hasChild} . \neg \text{Person})) (\text{PETER})$

$((\forall \text{hasChild} . \neg \text{Person})) (\text{PAUL})$

règle- \forall \mathcal{A} contient $(\forall R.C)(x)$ et $R(x,y)$, mais ne
contient pas $C(y)$

On ajoute $C(y)$ à \mathcal{A}

$\neg \text{Person}(\text{HARRY})$ ferme car $\text{Person}(\text{HARRY})$

Conclusion

- DL est un formalisme de représentation de connaissance combinant les graphes sémantiques et la logique
 - inférences sont faites par la *subsumption*
- DL est indépendant du web sémantique
- DL a servi de base de formalisation pour OWL

Bibliographie

- Michel Gagnon, Logique descriptive et OWL, 2012.
- P. Fournier-Viger, Une introduction aux logiques de description, 2011.
- P. Hitzler, M. Krötzsch, S. Rudolph, Foundations of Semantic Web, CRC Press, 2010, chap 5.
- F. Baader, W. Nutt, Basic Description Logics, Chap 2 du Description Logics Handbook
- Site de ressources sur la Logique de description: <http://dl.kr.org/>
- Implantations de systèmes d'inférence DL: <http://www.cs.man.ac.uk/~sattler/reasoners.html>