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

rali

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

rali

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 ⊑ Person`
  - `Mother ≡ Woman ⊓ ∃hasChild.Person`
  - `hasChild(MARY,PETER)`
  - `Woman(MARY)`
  - `Person(PETER)`
- Déduire de nouveaux faits
  - `Person(MARY)`
  - `Mother(MARY)`

rali

Mécanismes d'inférence

- Soit les définitions suivantes
  - $\texttt{Woman} \sqsubseteq \texttt{Person}$
  - $\texttt{Mother} \equiv \texttt{Woman} \sqcap \exists\texttt{hasChild.Person}$

  TBox
  - `hasChild(MARY,PETER)`
  - `Woman(MARY)`
  - `Person(PETER)`
- Déduire de nouveaux faits
  - `Person(MARY)`
  - `Mother(MARY)`

Mécanismes d'inférence

- Soit les définitions suivantes
  - `Woman ⊑ Person`
  - `Mother ≡ Woman ⊓ ∃hasChild.Person`

  TBox

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

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

rali

Inférences DL - 1
TBox: axiomes sur les concepts

- Un concept est-il consistant ?
  - peut-il exister au moins une instance d'une classe ?
- *Subsomption*: un concept peut-il être déduit?
  - `Woman ⊑ Person`
  - `Person ⊑ Animal`

  $\Rightarrow$ `Woman ⊑ 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

$$
\begin{aligned}
\mathtt{Woman} &\equiv \mathtt{Person} \sqcap \mathtt{Female}\\
\mathtt{Man} &\equiv \mathtt{Person} \sqcap \neg\mathtt{Woman}\\
\mathtt{Mother} &\equiv \mathtt{Woman} \sqcap \exists\mathtt{hasChild.Person}\\
\mathtt{Father} &\equiv \mathtt{Man} \sqcap \exists\mathtt{hasChild.Person}\\
\mathtt{Parent} &\equiv \mathtt{Father} \sqcup \mathtt{Mother}\\
\mathtt{Grandmother} &\equiv \mathtt{Mother} \sqcap \exists\mathtt{hasChild.Parent}\\
\mathtt{MotherWithManyChildren} &\equiv \mathtt{Mother} \sqcap {\geq}3\ \mathtt{hasChild}\\
\mathtt{MotherWithoutDaughter} &\equiv \mathtt{Mother} \sqcap \forall\mathtt{hasChild}.\neg\mathtt{Woman}\\
\mathtt{Wife} &\equiv \mathtt{Woman} \sqcap \exists\mathtt{hasHusband.Man}
\end{aligned}
$$

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

TBox

$$
\begin{aligned}
\texttt{Woman} &\equiv \texttt{Person} \sqcap \texttt{Female}\\
\texttt{Man} &\equiv \texttt{Person} \sqcap \neg\texttt{Woman}\\
\texttt{Mother} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasChild.Person}\\
\texttt{Father} &\equiv \texttt{Man} \sqcap \exists\texttt{hasChild.Person}\\
\texttt{Parent} &\equiv \texttt{Father} \sqcup \texttt{Mother}\\
\texttt{Grandmother} &\equiv \texttt{Mother} \sqcap \exists\texttt{hasChild.Parent}\\
\texttt{MotherWithManyChildren} &\equiv \texttt{Mother} \sqcap \geq 3\ \texttt{hasChild}\\
\texttt{MotherWithoutDaughter} &\equiv \texttt{Mother} \sqcap \forall\texttt{hasChild.}\neg\texttt{Woman}\\
\texttt{Wife} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasHusband.Man}
\end{aligned}
$$

ABox

MotherWithoutDaughter(MARY)
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 :

  $C, D ::= A \mid \bot \mid \top \mid \neg A \mid C \sqcap D \mid \forall R.C \mid \exists R.\top$
  - négation que de concepts atomiques
  - seul $\top$ est utilisable dans l'existence
- traduction possible en logique des prédicats

rali

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

rali

Interprétation par des ensembles

Δ^I = domaine d'interprétation

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

$\top^I = \Delta^I$

$\bot^I = \emptyset^I$

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

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

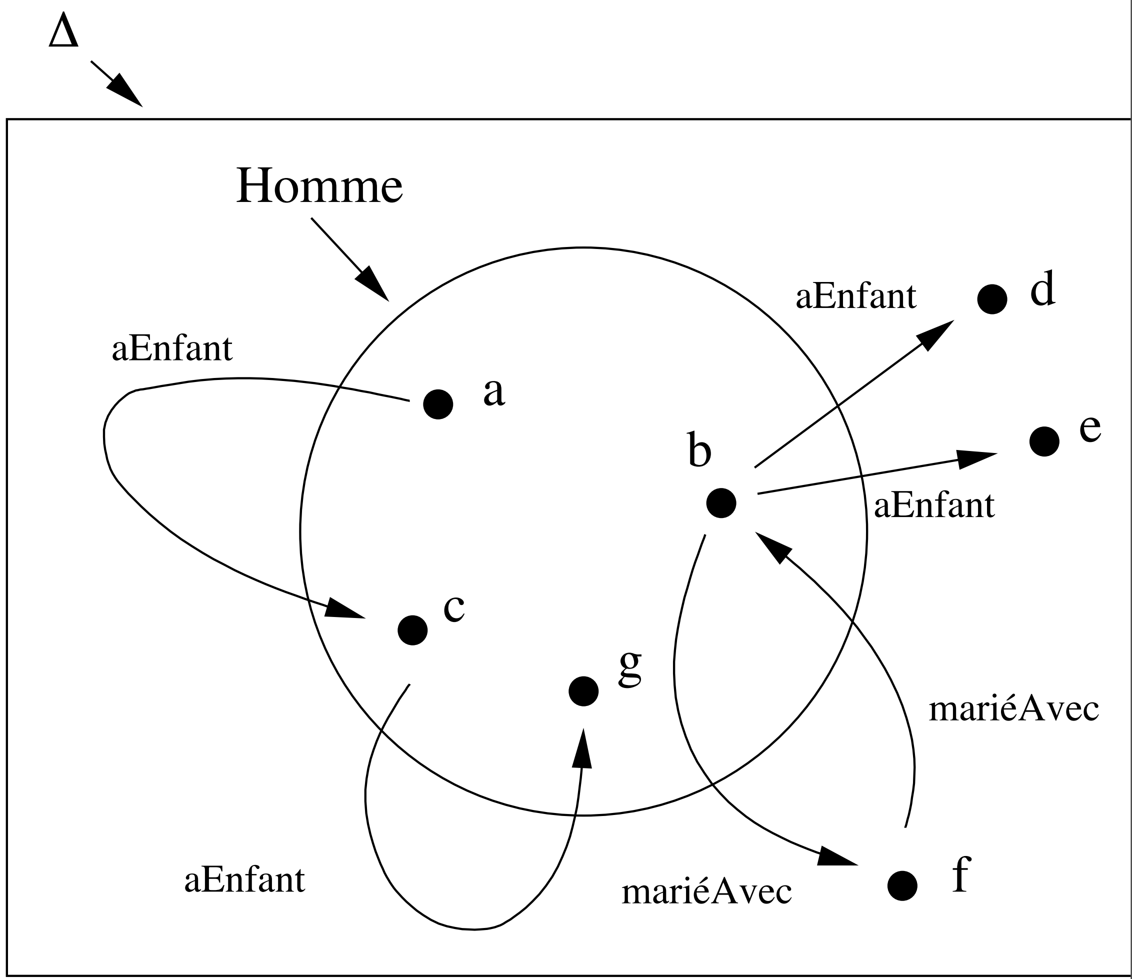
$(\forall R.C)^I = \{a \in \Delta^I \mid \forall b.(a,b) \in R^I \rightarrow b \in C^I\}$

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

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

Exemple d'interprétation

Parent $\equiv \exists$aEnfant.$\top$
ParentDeFemme $\equiv \exists$aEnfant.$\top \sqcap \forall$aEnfant.$\neg$Homme
Célibataire $\equiv \forall$mariéAvec.$\bot$
HommeMarié $\equiv$ Homme $\sqcap \exists$mariéAvec.$\top$
GrandParentChoyé $\equiv \forall$aEnfant.($\exists$aEnfant.$\top$)



Parent : $\{a, b, c\}$
ParentDeFemme : $\{b\}$
Célibataire : $\{a, c, d, e, g\}$
HommeMarié : $\{b\}$
GrandParentChoyé : $\{a, d, e, f, g\}$

$\Delta = \{a, b, c, d, e, f, g\}$
$\mathcal{I}$(Homme) $= \{a, b, c, g\}$
$\mathcal{I}$(aEnfant) $= \{(a, c), (b, d), (b, e), (c, g)\}$
$\mathcal{I}$(mariéAvec) $= \{(b, f), (f, b)\}$

Exemple tiré de : Michel Gagnon, *Logique descriptive et OWL*, École polytechnique, Montréal, 2012.

Autre interprétation possible!

Parent $\equiv \exists$aEnfant.$\top$
ParentDeFemme $\equiv \exists$aEnfant.$\top \sqcap \forall$aEnfant.$\neg$Homme
Célibataire $\equiv \forall$mariéAvec.$\bot$
HommeMarié $\equiv$ Homme $\sqcap \exists$mariéAvec.$\top$
GrandParentChoyé $\equiv \forall$aEnfant.($\exists$aEnfant.$\top$)

$$\Delta = \{a\}$$
$$\mathcal{I}(\text{Homme}) = \{\}$$
$$\mathcal{I}(\text{aEnfant}) = \{\}$$
$$\mathcal{I}(\text{mariéAvec}) = \{\}$$

Exemple tiré de : Michel Gagnon, *Logique descriptive et OWL*, École polytechnique, Montréal, 2012.

rali

Familles $\mathcal{AL}$ - 1

$\mathcal{C}$: négations de concepts arbitraires

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

$\mathcal{U}$: Disjonction

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

$\mathcal{E}$: quantification existentielle complète

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

$\mathcal{F}$: fonction plutôt que relation (p.e. `mariéAvec`)

$\mathcal{I}$: inversion (p.e. `enfantDe ≡ parentDe⁻`)

rali

Familles $\mathcal{AL}$ - 2

$\mathcal{N}$: Restriction de cardinalité

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

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

p.e. `Mother ⊓ ≥3hasChild`

$\mathcal{Q}$: Restriction de cardinalité qualifiée

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

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

p.e. `Mother ⊓ ≥3 hasChild.Woman`

rali

Familles $\mathcal{AL}$ - 3

$\mathcal{O}$: Nominaux : énumération d'individus

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

$\mathcal{R}$: 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

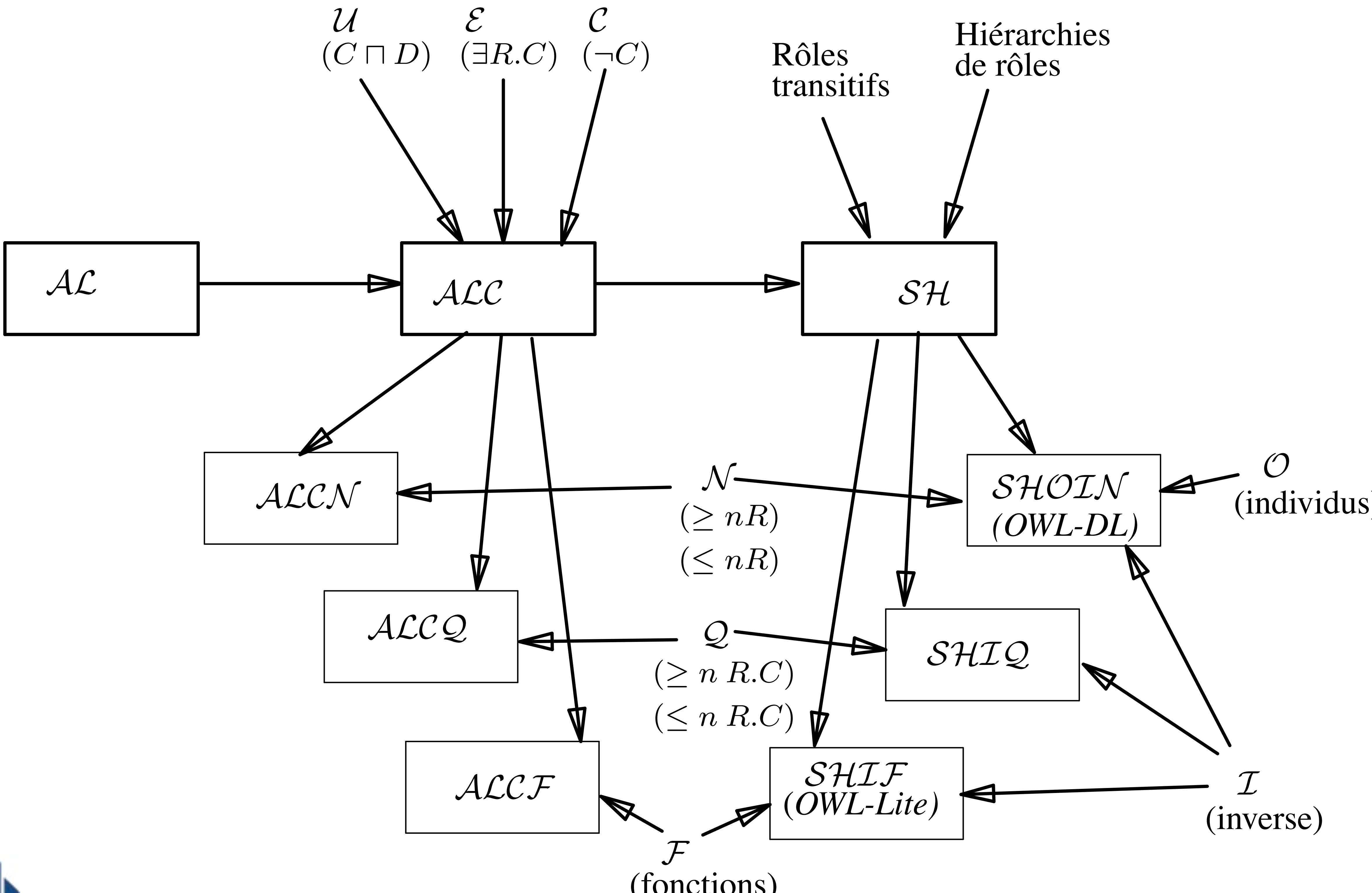


Figure tirée de : Michel Gagnon, *Logique descriptive et OWL*, École polytechnique, Montréal, 2012.

Constructions OWL en $\mathcal{ALC}$

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

- 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
D	types de données

OWL vs DL

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

Navigateur de complexité de DL

rali

Expression du nouvel $\mathcal{AL}$

Expression du nouvel $\mathcal{AL}$

$$
\begin{aligned}
\text{Person} \sqcap (&\leqq_1 \text{hasChild} \sqcup \\
&(\geqq_3 \text{hasChild} \sqcap \exists\text{hasChild.Female}))
\end{aligned}
$$

Expression du nouvel $\mathcal{AL}$

Person $\sqcap$ ($\leqq_1$ hasChild $\sqcup$
($\geqq_3$ 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 - 1

Fig 5.7 de Hitzler et al.,2010

concept ≡ prédicat unaire avec variable libre

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

rôle ≡ prédicat primitif binaire

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

inclusion de classes

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

$\sqcap$, $\sqcup$, $\neg$ ≡ 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

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

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

$\pi_x(\geq nS.C) =$

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

$\pi_x(\leq nS.C) =$

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

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

DL peut aussi être traduit en logique des prédicats - 3

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

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

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

$\pi_{x,y}(R_1 \bullet \ldots \bullet R_n) =$

$\quad \exists x_1 \ldots x_{n-1} \ (\pi_{x,x1}(R_1) \wedge \Lambda_{i=1..n-2} (R_{i+1}) \wedge \pi_{xn-1,y}(R_n)$

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

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

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

DL peut aussi être traduit en logique des prédicats - 3

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

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

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

$\pi_{x,y}(R_1 \bullet \ldots \bullet R_n) =$

$\quad \exists x_1 \ldots x_{n-1} \; (\pi_{x,x1}(R_1) \wedge \Lambda_{i=1..n-2} \, (R_{i+1}) \wedge \pi_{xn-1,y}(R_n)$

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

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

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

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 -1

Fig 5.9 de Hitzler et al.,2010

- $\text{Professor} \sqsubseteq \text{FacultyMember}$

  $\forall x(\mathit{Professor}(x) \rightarrow \mathit{FacultyMember}(x))$
- $\text{Professor} \sqsubseteq (\text{Person} \sqcap \text{FacultyMember})$

  $\sqcup\ (\text{Person} \sqcap \neg\text{PhDStudent})$

  $\forall x(\mathit{Professor}(x) \rightarrow (\mathit{Person}(x) \land \mathit{FacultyMember}(x))$

  $\lor\ (\mathit{Person}(x) \land \neg\mathit{PhDStudent}(x)))$
- $\text{Exam} \sqsubseteq \forall \text{hasExaminer.Professor}$

  $\forall x\ (\text{Exam}(x) \rightarrow \forall y(\text{hasExaminer}(x,y) \rightarrow \text{Professor}(y)))$

Exemples de traductions DL en logique des prédicats - 2

- Exam $\sqsubseteq$ $\leq$2hasExaminer
  $\forall x\ (Exam(x) \rightarrow \neg\exists_{x1,x2,x3}\ (x_1 \neq x_2) \wedge (x_2 \neq x_3) \wedge (x_1 \neq x_3)$
  $\wedge\ hasExaminer(x,x_1) \wedge hasExaminer(x,x_2)$
  $\wedge\ hasExaminer(x,x_3))$
- hasParent • hasBrother $\sqsubseteq$ hasUncle
  $\forall x \forall y (\exists x'\ hasParent(x,x') \wedge hasBrother(x',y)$
  $\rightarrow hasUncle(x,y))$

Inférences avec TBox

- Satisfiabilité : existe-il un modèle pour C ?
- Subsomption 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 ($C \sqcap D \sqsubseteq \bot$)

Inférences avec TBox

- Satisfiabilité : existe-il un modèle pour C ?
- Subsomption 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 insatifiabilité, 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

rali

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

rali

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 *Negative Normal Form* (NNF)
  - montrer que la négation de ce qu'on veut prouver est insatisfaisable

rali

Transformation en NNF ($\mathcal{N}$)

Fig 5.10 de Hitzler et al.,2010

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

$$\mathcal{N}(\forall R.C) \Rightarrow \forall R.\mathcal{N}(C)$$
$$\mathcal{N}(\exists R.C) \Rightarrow \exists R.\mathcal{N}(C)$$
$$\mathcal{N}(\neg\forall R.C) \Rightarrow \exists R.\mathcal{N}(\neg C)$$
$$\mathcal{N}(\neg\exists R.C) \Rightarrow \forall R.\mathcal{N}(\neg C)$$
$$\mathcal{N}(\leq nR.C) \Rightarrow \leq nR.\mathcal{N}(C)$$
$$\mathcal{N}(\geq nR.C) \Rightarrow \geq nR.\mathcal{N}(C)$$
$$\mathcal{N}(\neg\leq nR.C) \Rightarrow \geq (n+1)R.\mathcal{N}(C)$$
$$\mathcal{N}(\neg\geq (n+1)R.C) \Rightarrow \leq nR.\mathcal{N}(C)$$
$$\mathcal{N}(\neg\geq 0R.C) \Rightarrow \bot$$

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 : $C_1(x)$ $C_2(x)$
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 $z_1, \ldots, z_n$ 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 $y_1, \ldots, y_n$, 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 $R(x, y_1), \ldots, R(x, y_{n+1})$. 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 $y_i \neq y_j$.

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

Exemple de preuve par tableau

TBox

$$\begin{aligned}
\texttt{Woman} &\equiv \texttt{Person} \sqcap \texttt{Female}\\
\texttt{Man} &\equiv \texttt{Person} \sqcap \neg\texttt{Woman}\\
\texttt{Mother} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasChild.Person}\\
\texttt{Father} &\equiv \texttt{Man} \sqcap \exists\texttt{hasChild.Person}\\
\texttt{Parent} &\equiv \texttt{Father} \sqcup \texttt{Mother}\\
\texttt{Grandmother} &\equiv \texttt{Mother} \sqcap \exists\texttt{hasChild.Parent}\\
\texttt{MotherWithManyChildren} &\equiv \texttt{Mother} \sqcap \geq 3\ \texttt{hasChild}\\
\texttt{MotherWithoutDaughter} &\equiv \texttt{Mother} \sqcap \forall\texttt{hasChild.}\neg\texttt{Woman}\\
\texttt{Wife} &\equiv \texttt{Woman} \sqcap \exists\texttt{hasHusband.Man}
\end{aligned}$$

ABox

```
MotherWithoutDaughter(MARY)
Father(PETER)
hasChild(MARY,PETER)
hasChild(MARY,PAUL)
hasChild(PETER,HARRY)
```

$\Rightarrow$ `Grandmother(MARY)` *?*

rali

Remplacement des définitions de la TBox

en absence de définitions circulaires

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

$$
\begin{aligned}
\mathtt{Woman} &\equiv \mathtt{Person} \sqcap \mathtt{Female} \\
\mathtt{Man} &\equiv \mathtt{Person} \sqcap \neg(\mathtt{Person} \sqcap \mathtt{Female}) \\
\mathtt{Mother} &\equiv (\mathtt{Person} \sqcap \mathtt{Female}) \sqcap \exists\mathtt{hasChild.Person} \\
\mathtt{Father} &\equiv (\mathtt{Person} \sqcap \neg(\mathtt{Person} \sqcap \mathtt{Female})) \sqcap \exists\mathtt{hasChild.Person} \\
\mathtt{Parent} &\equiv ((\mathtt{Person} \sqcap \neg(\mathtt{Person} \sqcap \mathtt{Female})) \sqcap \exists\mathtt{hasChild.Person}) \\
&\quad \sqcup ((\mathtt{Person} \sqcap \mathtt{Female}) \sqcap \exists\mathtt{hasChild.Person}) \\
\mathtt{Grandmother} &\equiv ((\mathtt{Person} \sqcap \mathtt{Female}) \sqcap \exists\mathtt{hasChild.Person}) \sqcap \\
&\quad \exists\mathtt{hasChild.}(((\mathtt{Person} \sqcap \neg(\mathtt{Person} \sqcap \mathtt{Female})) \sqcap \\
&\qquad \exists\mathtt{hasChild.Person}) \sqcup \\
&\qquad ((\mathtt{Person} \sqcap \mathtt{Female}) \sqcap \exists\mathtt{hasChild.Person})) \\
\mathtt{MotherWithManyChildren} &\equiv ((\mathtt{Person} \sqcap \mathtt{Female}) \sqcap \exists\mathtt{hasChild.Person}) \sqcap \\
&\qquad \geq 3\ \mathtt{hasChild} \\
\mathtt{MotherWithoutDaughter} &\equiv \mathtt{Person} \sqcap \mathtt{Female} \sqcap \\
&\quad \forall\mathtt{hasChild.}(\neg\mathtt{Person} \sqcup \neg\mathtt{Female})
\end{aligned}
$$

Élimination de la TBox

en absence de définitions circulaires

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

$$\mathtt{Woman} \equiv \mathtt{Person} \sqcap \mathtt{Female}$$

$$\mathtt{Man} \equiv \mathtt{Person} \sqcap \neg\mathtt{Female}$$

$$\mathtt{Mother} \equiv \mathtt{Person} \sqcap \mathtt{Female} \sqcap \exists\mathtt{hasChild.Person}$$

$$\mathtt{Father} \equiv \mathtt{Person} \sqcap \neg\mathtt{Female} \sqcap \exists\mathtt{hasChild.Person}$$

$$\mathtt{Parent} \equiv \mathtt{Person} \sqcap \exists\mathtt{hasChild.Person}$$

$$\mathtt{Grandmother} \equiv \mathtt{Person} \sqcap \mathtt{Female} \sqcap \exists\mathtt{hasChild.Person} \sqcap \exists\mathtt{hasChild.}(\mathtt{Person} \sqcap \exists\mathtt{hasChild.Person})$$

$$\mathtt{MotherWithManyChildren} \equiv \mathtt{Person} \sqcap \mathtt{Female} \sqcap \geq 3\ \mathtt{hasChild}$$

$$\mathtt{MotherWithoutDaughter} \equiv \mathtt{Person} \sqcap \mathtt{Female} \sqcap \forall\mathtt{hasChild.}(\neg\mathtt{Person} \sqcup \neg\mathtt{Female})$$

rali

Opérations sur la ABox

```
Person(MARY)
Female(MARY)
∃hasChild.Person(MARY)
Person(PETER)
¬Female(PETER)
hasChild(PETER,w). Person(w)
hasChild(MARY,PETER)
hasChild(MARY,PAUL)
hasChild(PETER,HARRY)
```

Règle ∃

Opérations sur la ABox

```
Person(MARY)
Female(MARY)
∃hasChild.Person(MARY)
Person(PETER)
¬Female(PETER)
hasChild(PETER,w). Person(w)
hasChild(MARY,PETER)
hasChild(MARY,PAUL)
hasChild(PETER,HARRY)
```

Règle ∃

Ajout de la négation de ce qu'on veut prouver

```
¬Grandmother(MARY)
```

```
¬Person(MARY) ⊔ ¬Female(MARY) ⊔
¬∃hasChild.Person(MARY) ⊔
¬(∃hasChild.(Person ⊓ ∃hasChild.Person))(MARY)
```

Opérations sur la ABox

Person(MARY) Fermer branche 1
Female(MARY)
∃hasChild.Person(MARY)
Person(PETER)
¬Female(PETER)
hasChild(PETER,w). Person(w) Règle ∃
hasChild(MARY,PETER)
hasChild(MARY,PAUL)
hasChild(PETER,HARRY)

Ajout de la négation de ce qu'on veut prouver

¬Grandmother(MARY)

¬Person(MARY) ⊔ ¬Female(MARY) ⊔
¬∃hasChild.Person(MARY) ⊔
¬(∃hasChild.(Person ⊓ ∃hasChild.Person))(MARY)

rali

Opérations sur la ABox

Person(MARY) ← Fermer branche 1

Female(MARY) ← Fermer branche2

∃hasChild.Person(MARY)

Person(PETER)

¬Female(PETER)

hasChild(PETER,w). Person(w) Règle ∃

hasChild(MARY,PETER)

hasChild(MARY,PAUL)

hasChild(PETER,HARRY)

Ajout de la négation de ce qu'on veut prouver

¬Grandmother(MARY)

¬Person(MARY) ⊔ ¬Female(MARY) ⊔

¬∃hasChild.Person(MARY) ⊔

¬(∃hasChild.(Person ⊓ ∃hasChild.Person))(MARY)

Fermer la troisième branche

```
¬∃hasChild.Person(MARY)

 ∀hasChild.¬Person(MARY)
```

Fermer la troisième branche

$\neg\exists$hasChild.Person(MARY)

$\forall$hasChild.$\neg$Person(MARY)

$\neg$Person(Paul)

$\neg$Person(Peter)

Règle $\forall$
ferme cette sous-branche

Fermer la quatrième branche

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

rali

Fermer la quatrième branche

$\neg(\exists hasChild.(Person \sqcap \exists hasChild.Person))(MARY)$

$(\forall hasChild.(\neg(Person \sqcap \exists hasChild.Person)))(MARY)$

rali

Fermer la quatrième branche

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

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

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

Fermer la quatrième branche

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

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

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

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

Fermer la quatrième branche

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

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

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

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

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

rali

Fermer la quatrième branche

$\neg(\exists hasChild.(Person \sqcap \exists hasChild.Person))(MARY)$

$(\forall hasChild.(\neg(Person \sqcap \exists hasChild.Person)))(MARY)$

$(\forall hasChild.(\neg Person \sqcup \neg\exists hasChild.Person)))(MARY)$

$(\forall hasChild.(\neg Person \sqcup \forall hasChild.\neg Person)))(MARY)$

$(\forall hasChild.(\neg Person))(MARY) \sqcup$
$((\forall hasChild.(\forall hasChild.\neg Person))(MARY)$

$\neg Person(Paul)$ Règle $\forall$

$\neg Person(Peter)$ ferme cette sous-branche

Fermer la 2e sous-branche

((∀hasChild.(∀hasChild.¬Person))(MARY)

((∀hasChild.¬Person))(PETER)

((∀hasChild.¬Person))(PAUL)

¬Person(HARRY) peut fermer si monde clos

Conclusion

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

rali

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

rali