

## Introduction : mots-clés et sigles

- DL = *description logic* = logique de description

(avant : logiques terminologiques)

‘description’ ⇒ **représenter** les connaissances

but : exprimer plus qu’en logique propositionnelle

‘logique’ ⇒ **raisonner** à partir de ces connaissances

but : avoir de meilleures propriétés calculatoires

que la logique des prédicats

- KB = *knowledge base* = base de connaissances

$$KB = TBox \cup ABox$$

- ABox = *assertion box*

= connaissances contingentes

- TBox = *terminological box*

= ontologie

= connaissances non contingentes

= axiomes

≈ contraintes d’intégrité

## Introduction : mots-clés et sigles

- DL = *description logic* = logique de description

(avant : logiques terminologiques)

‘description’ ⇒ **représenter** les connaissances

but : exprimer plus qu’en logique propositionnelle

‘logique’ ⇒ **raisonner** à partir de ces connaissances

but : avoir de meilleures propriétés calculatoires

que la logique des prédicats

- KB = *knowledge base* = base de connaissances

$$KB = TBox \cup ABox$$

- ABox = *assertion box*

= **connaissances contingentes**

- TBox = *terminological box*

= ontologie

= **connaissances non contingentes**

= **axiomes**

≈ contraintes d’intégrité

## Introduction : la ABox

ABox = *assertion box*

- vraies à un instant donné dans le temps (qui reste implicite)
  - ① prédicats unaires : propriétés d'individus

Feminine(Anne)

Masculin(Charles)

- ② prédicats binaires : relations entre individus

mereDe(Anne, Charles)

pereDe(Bob, Charles)

pereDe(Charles, Dan)

⇒ la ABox utilise les axiomes de la TBox ...

## Introduction : la ABox

ABox = *assertion box*

- vraies à un instant donné dans le temps (qui reste implicite)
  - ① prédicats unaires : propriétés d'individus

Feminine(Anne)

Masculin(Charles)

- ② prédicats binaires : relations entre individus

mereDe(Anne, Charles)

pereDe(Bob, Charles)

pereDe(Charles, Dan)

⇒ la ABox utilise les axiomes de la TBox ...

## Introduction : la TBox

TBox = *terminological box*

- axiomatise les propriétés et les relations
  - terminologie DL: concepts et rôles
  - principalement : inclusion de concepts

Mere  $\sqsubseteq$  Parent

Mere  $\sqsubseteq$  Parent  $\sqcap$  Feminine

Mere  $\sqsubseteq$   $\exists$ mereDe.T

$\exists$ mereDe.T  $\sqsubseteq$  Mere

$\exists$ mereDe.( $\exists$ mereDe.T)  $\sqsubseteq$  GrandMere

- aussi: équivalence de concepts

Mere  $\equiv$   $\exists$ mereDe.T

- et parfois aussi: inclusion de rôles

mereDe  $\sqsubseteq$  parentDe

- vrais pour tout instant temporel (qui reste implicite)

## Introduction : la TBox

TBox = *terminological box*

- axiomatise les propriétés et les relations
  - terminologie DL: concepts et rôles
  - principalement : inclusion de concepts

Mere  $\sqsubseteq$  Parent

Mere  $\sqsubseteq$  Parent  $\sqcap$  Feminine

Mere  $\sqsubseteq$   $\exists$ mereDe.T

$\exists$ mereDe.T  $\sqsubseteq$  Mere

$\exists$ mereDe.( $\exists$ mereDe.T)  $\sqsubseteq$  GrandMere

- aussi: équivalence de concepts

Mere  $\equiv$   $\exists$ mereDe.T

- et parfois aussi: inclusion de rôles

mereDe  $\sqsubseteq$  parentDe

- vrais pour tout instant temporel (qui reste implicite)

## Introduction : les concepts

un concept dénote un ensemble d'individus

- concept atomique : Parent, Masculin, Feminine, Mere, ...
  - ① concept atomique primitif : Parent, Masculin, ...
  - ② concept atomique non-primitif : Mere, GrandMere, ...  
⇒ définis dans la TBox en termes d'autres concepts :

$$\text{Femme} \equiv \text{Personne} \sqcap \text{Feminine}$$

$$\text{Mere} \equiv \exists \text{mereDe}.\top$$

- concept complexe :  $\text{Personne} \sqcap \text{Feminine}$ ,  $\exists \text{mereDe}.\top$ ,  $\exists \text{mereDe}.\left(\exists \text{mereDe}.\top\right)$ , ...
  - constructeurs de concepts complexes :
    - ① opérateurs booléens :  $\neg C$ ,  $C \sqcap C$ ,  $C \sqcup C$
    - ② opérateurs combinant concepts et rôles :  $\exists R.C$ ,  $\forall R.C$

## Introduction : les rôles

un rôle dénote un ensemble de couples d'individus

- rôle atomique :
  - `mereDe`, `partieDe`, `mange`, ...
- rôle complexe : `pereDe`  $\sqcap$  `grandPereDe`
- constructeurs de rôles complexes:
  - ① opérateurs booléens :  $\neg R$ ,  $R \sqcap R$ ,  $R \sqcup R$
  - ② opérateurs de l'algèbre des relations :
    - $R \circ R$  (composition)
    - $R^*$  (itération, *Kleene star*)
    - $R^{-1}$  (converse)
    - $\bar{R}$  (complément)

## Introduction : l'aspect raisonnement

raisonner = 'extraire de la KB plus que ce qu'il y a écrit'

- exemple :  $KB = \mathcal{T} \cup \mathcal{A}$ , où

$$\mathcal{A} = \{\text{Pere}(\text{Bob})\}$$

$$\mathcal{T} = \{\text{Pere} \sqsubseteq \text{Personne}\}$$

- KB ne contient pas  $\text{Personne}(\text{Bob})$
- KB a comme **conséquence logique**  $\text{Personne}(\text{Bob})$  !
- services de raisonnement :
  - 1 inférence de propriétés d'individus
  - 2 inférence de relations entre individus
  - 3 subsomption de concepts
  - 4 classification: calculer la hierarchie de subsomption entre concepts
  - 5 satisfaisabilité
  - 6 non-vacuité de concepts (pas de A tel que  $KB \models A \equiv \perp$ )
  - 7 non-redondance de concepts (pas de A, A' t.q.  $KB \models A \equiv A'$ )

## Introduction : l'aspect raisonnement

raisonner = 'extraire de la KB plus que ce qu'il y a écrit'

- exemple :  $KB = \mathcal{T} \cup \mathcal{A}$ , où

$$\mathcal{A} = \{\text{Pere}(\text{Bob})\}$$

$$\mathcal{T} = \{\text{Pere} \sqsubseteq \text{Personne}\}$$

- KB ne contient pas  $\text{Personne}(\text{Bob})$
- KB a comme **conséquence logique**  $\text{Personne}(\text{Bob})$  !
- services de raisonnement :
  - 1 inférence de propriétés d'individus
  - 2 inférence de relations entre individus
  - 3 subsomption de concepts
  - 4 classification: calculer la hierarchie de subsomption entre concepts
  - 5 satisfaisabilité
  - 6 non-vacuité de concepts (pas de  $A$  tel que  $KB \models A \equiv \perp$ )
  - 7 non-redondance de concepts (pas de  $A, A'$  t.q.  $KB \models A \equiv A'$ )

# Introduction : avantages des DL

- 1 sémantiques claires et bien définies
  - ≠ langage naturel
- 2 tâches de raisonnement **décidables**
  - ≠ logique du premier ordre FOL
  - plus difficiles que pour la logique propositionnelle
    - raison : les DL sont plus expressifs
- 3 applications multiples
  - web sémantique
  - bases de données (modèles entité-relation exprimables en DL)
  - génie du logiciel (diagrammes UML exprimables en DL)
- 4 flexibles
  - hiérarchie des DL
    - chaque DL est déterminée par des restrictions du langage
  - expressivité vs. complexité

## Introduction : complexité vs. expressivité

- restrictions de l'expressivité :
  - pas de négation ' $\neg$ ' ; pas de disjonction ' $\sqcup$ ' ; ...
  - pas d'inclusion de rôles ' $\sqsubseteq$ ' ; pas de rôles complexes ; ...
  - dans  $\exists R.C$  il faut que  $C = \top$  ; ...
- expressivité du langage  $\Rightarrow$  complexité du raisonnement
  - PTIME = temps polynomial
  - NPTIME = temps polynomial non-déterministe
  - PSPACE = espace polynomial (PSPACE = NPSPACE)
  - EXPTIME = temps exponentiel
  - NEXPTIME = temps exponentiel non-déterministe
  - EXPSPACE = espace exponentiel

$\Rightarrow$  'navigateur de complexité' sur <http://dl.kr.org>