Ontologies et web sémantique:

Introduction aux logiques de description

Andreas Herzig
CNRS, IRIT, Univ. Toulouse
`http://www.irit.fr/~Andreas.Herzig`

(en partie repris du cours de Nicola Olivetti,
Univ. Aix-Marseille, `http://www.lsis.org/olivetti`)

Univ. Toulouse, M2 informatique
parcours *Données et connaissances*
2021-22

Introduction : resources

- livres :
  - Franz Baader, Diego Calvanese, Deborah L. McGuinness, Daniele Nardi and Peter F. Patel-Schneider (eds.)
    The Description Logic Handbook: Theory, Implementation, and Applications
    Cambridge University Press, 2003 (reédité en 2007 ; en ligne depuis 2010)
  - Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler
    An Introduction to Description Logic
    Cambridge University Press, 2017
    `http://dltextbook.org`
- *le* site web : `http://dl.kr.org`

$\Rightarrow$ DL workshop
$\Rightarrow$ conférences afférantes
$\Rightarrow$ entrepôt d'ontologies

Introduction : mots-clés et sigles

- DL = *description logic* = logique de description

  (avant : logiques terminologiques)

  'description' $\Rightarrow$ représenter les connaissances

  but : exprimer plus qu'en logique propositionnelle

  'logique' $\Rightarrow$ raisonner à partir de ces connaissances

  but : avoir de meilleures propriétés calculatoires que la logique des prédicats

- KB = *knowledge base* = base de connaissances

$$\boxed{\text{KB} = \text{TBox} \cup \text{ABox}}$$

- 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

$$\boxed{\text{KB} = \text{TBox} \cup \text{ABox}}$$

- ABox = *assertion box*

  = connaissances contingentes

- TBox = *terminological box*

  = ontologie

  = connaissances non contingentes

  = axiomes

  ≈ contraintes d'intégrité

Introduction : la ABox

$\boxed{\text{ABox} = \textit{assertion box}}$

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

$$\texttt{Feminine}(\texttt{Anne})$$
$$\texttt{Masculin}(\texttt{Charles})$$

  2. prédicats binaires : relations entre individus

$$\texttt{mereDe}(\texttt{Anne}, \texttt{Charles})$$
$$\texttt{pereDe}(\texttt{Bob}, \texttt{Charles})$$
$$\texttt{pereDe}(\texttt{Charles}, \texttt{Dan})$$

⇒ 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

$$\mathtt{Feminine}(\mathtt{Anne})$$
$$\mathtt{Masculin}(\mathtt{Charles})$$

  2. prédicats binaires : relations entre individus

$$\mathtt{mereDe}(\mathtt{Anne}, \mathtt{Charles})$$
$$\mathtt{pereDe}(\mathtt{Bob}, \mathtt{Charles})$$
$$\mathtt{pereDe}(\mathtt{Charles}, \mathtt{Dan})$$

$\Rightarrow$ 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

$$\mathtt{Mere} \sqsubseteq \mathtt{Parent}$$

$$\mathtt{Mere} \sqsubseteq \mathtt{Parent} \sqcap \mathtt{Feminine}$$

$$\mathtt{Mere} \sqsubseteq \exists \mathtt{mereDe}.\top$$

$$\exists \mathtt{mereDe}.\top \sqsubseteq \mathtt{Mere}$$

$$\exists \mathtt{mereDe}.(\exists \mathtt{mereDe}.\top) \sqsubseteq \mathtt{GrandMere}$$

  - aussi: équivalence de concepts

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

  - et parfois aussi: inclusion de rôles

$$\mathtt{mereDe} \sqsubseteq \mathtt{parentDe}$$

- 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

$$\mathtt{Mere} \sqsubseteq \mathtt{Parent}$$
$$\mathtt{Mere} \sqsubseteq \mathtt{Parent} \sqcap \mathtt{Feminine}$$
$$\mathtt{Mere} \sqsubseteq \exists \mathtt{mereDe}.\top$$
$$\exists \mathtt{mereDe}.\top \sqsubseteq \mathtt{Mere}$$
$$\exists \mathtt{mereDe}.(\exists \mathtt{mereDe}.\top) \sqsubseteq \mathtt{GrandMere}$$

  - aussi: équivalence de concepts

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

  - et parfois aussi: inclusion de rôles

$$\mathtt{mereDe} \sqsubseteq \mathtt{parentDe}$$

- vrais pour tout instant temporel (qui reste implicite)

Introduction : les concepts

$\boxed{\text{un concept dénote un ensemble d'individus}}$

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

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

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

- concept complexe : $\mathtt{Personne} \sqcap \mathtt{Feminine}$, $\exists \mathtt{mereDe}.\top$, $\exists \mathtt{mereDe}.(\exists \mathtt{mereDe}.\top)$,...
  - 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 : $\neg \mathrm{R}$, $\mathrm{R} \sqcap \mathrm{R}$, $\mathrm{R} \sqcup \mathrm{R}$
  2. opérateurs de l'algèbre des relations :
     - $\mathrm{R} \circ \mathrm{R}$ (composition)
     - R^* (itération, *Kleene star*)
     - R^{-1} (converse)
     - $\bar{\mathrm{R}}$ (complément)

Introduction : l'aspect raisonnement

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

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

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

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

  - $\mathtt{KB}$ ne contient pas $\mathtt{Personne}(\mathtt{Bob})$
  - $\mathtt{KB}$ a comme conséquence logique $\mathtt{Personne}(\mathtt{Bob})$!

- 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 $\mathtt{A}$ tel que $KB \models \mathtt{A} \equiv \bot$)
  7. non-redondance de concepts (pas de $\mathtt{A}$, $\mathtt{A}'$ t.q. $KB \models \mathtt{A} \equiv \mathtt{A}'$)

Introduction : l'aspect raisonnement

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

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

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

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

  - $\mathtt{KB}$ ne contient pas $\mathtt{Personne}(\mathtt{Bob})$
  - $\mathtt{KB}$ a comme conséquence logique $\mathtt{Personne}(\mathtt{Bob})$!

- 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 $\mathtt{A}$ tel que $KB \models \mathtt{A} \equiv \bot$)
  7. non-redondance de concepts (pas de $\mathtt{A}$, $\mathtt{A}'$ t.q. $KB \models \mathtt{A} \equiv \mathtt{A}'$)

Introduction : avantages des DL

1. sémantiques claires et bien définies
   - $\neq$ langage naturel
2. tâches de raisonnement décidables
   - $\neq$ 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-détérministe
PSPACE	=	espace polynomial (PSPACE = NPSPACE)
EXPTIME	=	temps exponentiel
NEXPTIME	=	temps exponentiel non-détérministe
EXPSPACE	=	espace exponentiel

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

Outline

Syntaxe

- ALC = Attributive Concept Language with Complements [Schmidt-Schauß & Smolka, 1991]

  N.B. : plutôt que ‘langage’, devrait s’appeller *logique*

  (logique = langage + sémantique)

- convention :
  - $\mathtt{A}$, $\mathtt{A}_i$, . . . : concepts atomiques
    - dans les exemples : commence par majuscule (`Parent`, . . .)
  - $\mathtt{C}$, $\mathtt{D}$, $\mathtt{C}_i$, . . . : concepts arbitraires
  - $\mathtt{R}$, $\mathtt{R}_i$, . . . : rôles atomiques
    - dans les exemples : commence par minuscule (`parentDe`, . . .)
  - *pas de rôles complexes*

Syntaxe : concepts complexes

- constructeurs de concepts :

$\top$	concept universel	"tout"
$\bot$	concept vide	"rien"
$\neg \mathtt{C}$	complément (ou : négation)	"non $\mathtt{C}$"
$\mathtt{C} \sqcap \mathtt{D}$	intersection (ou : conjonction)	"$\mathtt{C}$ et $\mathtt{D}$"
$\mathtt{C} \sqcup \mathtt{D}$	union (ou : disjonction)	"$\mathtt{C}$ ou $\mathtt{D}$"
$\forall \mathtt{R}.\mathtt{C}$	quantification universelle restreinte	"tous les $\mathtt{R}$-successeurs sont dans $\mathtt{C}$"
$\exists \mathtt{R}.\mathtt{C}$	quantification existentielle restreinte	"il existe un $\mathtt{R}$-successeur qui est dans $\mathtt{C}$"

- grammaire des concepts ('BNF') :

$$\mathtt{C} ::= \mathrm{N_C} \mid \top \mid \bot \mid \neg \mathtt{C} \mid (\mathtt{C} \sqcap \mathtt{C}) \mid (\mathtt{C} \sqcup \mathtt{C}) \mid (\forall \mathrm{N_R}.\mathtt{C}) \mid (\exists \mathrm{N_R}.\mathtt{C})$$

où :

- $\mathtt{C}$: symbole non-terminal de la grammaire
- $\mathrm{N_C} \in$ *NomsConcepts*, $\mathrm{N_R} \in$ *NomsRoles* ('symboles terminaux')
- $\top, \bot, \neg, \sqcap, \sqcup$: connecteurs logiques

Syntaxe : concepts complexes

- constructeurs de concepts :

$\top$	concept universel	"tout"
$\bot$	concept vide	"rien"
$\neg C$	complément (ou : négation)	"non C"
$C \sqcap D$	intersection (ou : conjonction)	"C et D"
$C \sqcup D$	union (ou : disjonction)	"C ou D"
$\forall R.C$	quantification universelle restreinte	"tous les R-successeurs sont dans C"
$\exists R.C$	quantification existentielle restreinte	"il existe un R-successeur qui est dans C"

- grammaire des concepts ('BNF') :

$$C ::= N_C \mid \top \mid \bot \mid \neg C \mid (C \sqcap C) \mid (C \sqcup C) \mid (\forall N_R.C) \mid (\exists N_R.C)$$

où :

- C: symbole non-terminal de la grammaire
- $N_C \in$ *NomsConcepts*, $N_R \in$ *NomsRoles* ('symboles terminaux')
- $\top$, $\bot$, $\neg$, $\sqcap$, $\sqcup$: connecteurs logiques

Syntaxe : exemples

- concepts atomiques :
  $\mathtt{Personne}$, $\mathtt{Masculin}$, $\mathtt{Feminine}$, $\mathtt{Riche}$, ...
- rôles atomiques :
  $\mathtt{parentDe}$, $\mathtt{mereDe}$, $\mathtt{pereDe}$
- concepts complexes :
  1. $\mathtt{Personne} \sqcap \mathtt{Feminine}$
  2. $\mathtt{Personne} \sqcap \neg\mathtt{Feminine}$
  3. $\mathtt{Personne} \sqcap \exists\mathtt{parentDe}.\top$
  4. $\mathtt{Personne} \sqcap \forall\mathtt{parentDe}.\bot$
  5. $\mathtt{Personne} \sqcap \exists\mathtt{parentDe}.\top \sqcap \forall\mathtt{parentDe}.\mathtt{Feminine}$
  6. $\mathtt{Personne} \sqcap (\mathtt{Riche} \sqcup \exists\mathtt{parentDe}.\mathtt{Riche})$
  7. $\mathtt{Personne} \sqcap \exists\mathtt{parentDe}.\exists\mathtt{parentDe}.\top$

Syntaxe : exercices

- étant donné les ensembles

$$\mathit{NomsConcepts} = \{\mathtt{Personne}, \mathtt{Masculin}, \mathtt{Feminine}, \mathtt{Riche}\}$$
$$\mathit{NomsRoles} = \{\mathtt{parentDe}, \mathtt{s\oe urDe}, \mathtt{frereDe}\},$$

définir les concepts complexes suivants :

1. être une femme
2. être une mère
3. être une mère qui n'a que des garçons
4. être un oncle
5. être un grand-oncle
6. être quelqu'un qui a un neveu riche
7. être quelqu'un qui a un oncle riche ??
   ⇒ pas de converse dans ALC
8. être quelqu'un qui n'a qu'un seul enfant ??
   ⇒ pas de restriction de cardinalité dans ALC

Sémantique

- interprétation: donne du sens aux concepts atomiques et aux rôles atomiques (dans un contexte particulier)
- interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ telle que :
  - $\Delta^{\mathcal{I}}$ est un ensemble non-vide (domaine d'individus)
  - $(.)^{\mathcal{I}}$: *NomsConcepts* $\longrightarrow 2^{\Delta^{\mathcal{I}}}$ (interprétation des concepts)
  - $(.)^{\mathcal{I}}$: *NomsRoles* $\longrightarrow 2^{\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}}$ (interprétation des rôles)

  logique du premier ordre : il y faut en plus interpréter les prédicats 3-aires, 4-aires,. . .

- une interprétation $\mathcal{I}$:
  - donne du sens aux concepts et rôles atomiques

$$(\mathtt{A})^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}}$$
$$(\mathtt{R})^{\mathcal{I}} \subseteq \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}$$

  - reste à donner du sens aux concepts complexes. . .

Sémantique (suite)

- interprétation de concepts complexes :

$$
\begin{aligned}
\top^{\mathcal{I}} &= \Delta^{\mathcal{I}} \\
\bot^{\mathcal{I}} &= \emptyset \\
(\neg \mathtt{C})^{\mathcal{I}} &= \Delta^{\mathcal{I}} \setminus \mathtt{C}^{\mathcal{I}} \\
(\mathtt{C} \sqcap \mathtt{D})^{\mathcal{I}} &= \mathtt{C}^{\mathcal{I}} \cap \mathtt{D}^{\mathcal{I}} \\
(\mathtt{C} \sqcup \mathtt{D})^{\mathcal{I}} &= \mathtt{C}^{\mathcal{I}} \cup \mathtt{D}^{\mathcal{I}} \\
(\forall \mathtt{R}.\mathtt{C})^{\mathcal{I}} &= \left\{ a \in \Delta^{\mathcal{I}} \;:\; \text{pour tout } b \in \Delta^{\mathcal{I}}, \text{ si } (a, b) \in \mathtt{R}^{\mathcal{I}} \text{ alors } b \in \mathtt{C}^{\mathcal{I}} \right\} \\
(\exists \mathtt{R}.\mathtt{C})^{\mathcal{I}} &= \left\{ a \in \Delta^{\mathcal{I}} \;:\; \text{il existe } b \in \Delta^{\mathcal{I}} \text{ tel que } (a, b) \in \mathtt{R}^{\mathcal{I}} \text{ et } b \in \mathtt{C}^{\mathcal{I}} \right\}
\end{aligned}
$$

Sémantique : quelques équivalences

équivalences sémantiques :

$$
\begin{aligned}
(\neg\neg C)^{\mathcal{I}} &= C^{\mathcal{I}} \\
(\neg(C \sqcap D))^{\mathcal{I}} &= (\neg C \sqcup \neg D)^{\mathcal{I}} \\
(\neg(C \sqcup D))^{\mathcal{I}} &= (\neg C \sqcap \neg D)^{\mathcal{I}} \\
(\neg\forall R.C)^{\mathcal{I}} &= (\exists R.\neg C)^{\mathcal{I}} \\
(\neg\exists R.C)^{\mathcal{I}} &= (\forall R.\neg C)^{\mathcal{I}}
\end{aligned}
$$

Proposition (forme normale négative)

Tout concept C *peut être transformé en un concept* $nnf(C)$ *tel que*

1. *dans* $nnf(C)$*, la négation apparaît seulement devant des concepts atomiques,*
2. *pour toute interprétation* $\mathcal{I}$*,* $(nnf(C))^{\mathcal{I}} = C^{\mathcal{I}}$*.*

- exemple : $\neg\forall \mathtt{parentDe}.\bot$ (…il manque une équivalence)

Sémantique : quelques équivalences

équivalences sémantiques :

$$
\begin{aligned}
(\neg\neg \mathtt{C})^{\mathcal{I}} &= C^{\mathcal{I}} \\
(\neg(\mathtt{C} \sqcap \mathtt{D}))^{\mathcal{I}} &= (\neg\mathtt{C} \sqcup \neg\mathtt{D})^{\mathcal{I}} \\
(\neg(\mathtt{C} \sqcup \mathtt{D}))^{\mathcal{I}} &= (\neg\mathtt{C} \sqcap \neg\mathtt{D})^{\mathcal{I}} \\
(\neg\forall \mathtt{R}.\mathtt{C})^{\mathcal{I}} &= (\exists \mathtt{R}.\neg\mathtt{C})^{\mathcal{I}} \\
(\neg\exists \mathtt{R}.\mathtt{C})^{\mathcal{I}} &= (\forall \mathtt{R}.\neg\mathtt{C})^{\mathcal{I}}
\end{aligned}
$$

Proposition (forme normale négative)

Tout concept $\mathtt{C}$ peut être transformé en un concept $nnf(\mathtt{C})$ tel que

1. *dans $nnf(\mathtt{C})$, la négation apparaît seulement devant des concepts atomiques,*
2. *pour toute interprétation $\mathcal{I}$, $(nnf(\mathtt{C}))^{\mathcal{I}} = \mathtt{C}^{\mathcal{I}}$.*

- exemple : $\neg\forall \mathtt{parentDe}.\bot$ (. . . il manque une équivalence)

Sémantique : quelques équivalences

équivalences sémantiques :

$$(\neg\neg \mathtt{C})^{\mathcal{I}} = C^{\mathcal{I}}$$

$$(\neg(\mathtt{C} \sqcap \mathtt{D}))^{\mathcal{I}} = (\neg \mathtt{C} \sqcup \neg \mathtt{D})^{\mathcal{I}}$$

$$(\neg(\mathtt{C} \sqcup \mathtt{D}))^{\mathcal{I}} = (\neg \mathtt{C} \sqcap \neg \mathtt{D})^{\mathcal{I}}$$

$$(\neg\forall \mathrm{R}.\mathtt{C})^{\mathcal{I}} = (\exists \mathrm{R}.\neg \mathtt{C})^{\mathcal{I}}$$

$$(\neg\exists \mathrm{R}.\mathtt{C})^{\mathcal{I}} = (\forall \mathrm{R}.\neg \mathtt{C})^{\mathcal{I}}$$

Proposition (forme normale négative)

Tout concept $\mathtt{C}$ *peut être transformé en un concept nnf*$(\mathtt{C})$ *tel que*

1. *dans nnf*$(\mathtt{C})$*, la négation apparaît seulement devant des concepts atomiques,*
2. *pour toute interprétation* $\mathcal{I}$*,* $(nnf(\mathtt{C}))^{\mathcal{I}} = \mathtt{C}^{\mathcal{I}}$.

- exemple : $\neg\forall\mathtt{parentDe}.\bot$ (... il manque une équivalence)

Outline

Le langage ALCN : ALC + restrictions de cardinalité

- ALC plus un nouveau constructeurs de concept : restriction de cardinalité (‘*Number restriction*’)
- grammaire des concepts (‘BNF’) :

$$\mathtt{C} ::= \mathtt{N_C} \mid \top \mid \bot \mid \neg\mathtt{C} \mid \mathtt{C} \sqcap \mathtt{C} \mid \mathtt{C} \sqcup \mathtt{C} \mid \forall \mathtt{N_R}.\mathtt{C} \mid \exists \mathtt{N_R}.\mathtt{C} \mid \leq n\ \mathtt{N_R} \mid \geq n\ \mathtt{N_R}$$

  où :
  - $\mathtt{N_C} \in$ *NomsConcepts*
  - $\mathtt{N_R} \in$ *NomsRoles*
  - $n \geq 0$ est un entier naturel
- lecture :
  - $\leq n\ \mathtt{R}$ = ‘il y a au plus n R-successeurs’
  - $\geq n\ \mathtt{R}$ = ‘il y a au moins n R-successeurs’
- exercice :
  - définir le concept $=n\ \mathtt{R}$ (‘il y a exactement n R-successeurs’)

Le langage ALCN : ALC + restrictions de cardinalité

- ALC plus un nouveau constructeurs de concept : restriction de cardinalité (‘*Number restriction*’)
- grammaire des concepts (‘BNF’) :

  $$\mathtt{C} ::= \mathtt{N_C} \mid \top \mid \bot \mid \neg\mathtt{C} \mid \mathtt{C} \sqcap \mathtt{C} \mid \mathtt{C} \sqcup \mathtt{C} \mid \forall\mathtt{N_R}.\mathtt{C} \mid \exists\mathtt{N_R}.\mathtt{C} \mid \leq n\ \mathtt{N_R} \mid \geq n\ \mathtt{N_R}$$

  où :

  - $\mathtt{N_C} \in$ *NomsConcepts*
  - $\mathtt{N_R} \in$ *NomsRoles*
  - $n \geq 0$ est un entier naturel
- lecture :
  - $\leq n\ \mathtt{R}$ = ‘il y a au plus n $\mathtt{R}$-successeurs’
  - $\geq n\ \mathtt{R}$ = ‘il y a au moins n $\mathtt{R}$-successeurs’
- exercice :
  - définir le concept $=n\ \mathtt{R}$ (‘il y a exactement n $\mathtt{R}$-successeurs’)

ALCN : sémantique

- les mêmes interprétations que dans ALC :
  - $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$ telle que
    - $\Delta^{\mathcal{I}}$ est un ensemble non-vide (domaine)
    - $(.)^{\mathcal{I}}$: *NomsConcepts* $\longrightarrow 2^{\Delta^{\mathcal{I}}}$ (interprétation des concepts)
    - $(.)^{\mathcal{I}}$: *NomsRoles* $\longrightarrow 2^{\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}}$ (interprétation des rôles)
- interprétation des concepts complexes :

$$(\leq n\,\mathrm{R})^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \ : \ Card(\{b \in \Delta \ : \ (a, b) \in \mathrm{R}^{\mathcal{I}}\}) \leq n \right\}$$

$$(\geq n\,\mathrm{R})^{\mathcal{I}} = \ldots$$

- strictement plus expressif que ALC :
  les restrictions de cardinalité ne peuvent pas toutes être exprimées dans ALC

  preuve non triviale (utilise la *bisimilation* entre interpretations)

ALCN : exercices

- décrire les concepts suivants :
  1. être une personne avec au moins trois enfants
  2. être une personne avec exactement trois enfants
  3. être un cours avec au plus 15 participants dont tous sont des étudiants de master
     - concepts primitifs : `Cours`, `EtudiantEnMaster`
     - rôle primitif : `aParticipant`
  4. être une grand-mère avec une fille qui a deux fils comme seuls enfants
- étendre la définition de la forme normale négative aux restrictions de cardinalité
- est-ce qu'on peut décrire le concept 'être une mère qui a (au moins) deux fils' ?

ALCQ : restrictions de cardinalité *qualifiées*

- grammaire des concepts ('BNF') :

$$\mathtt{C} ::= \mathtt{N_C} \mid \top \mid \bot \mid \neg\mathtt{C} \mid \mathtt{C} \sqcap \mathtt{C} \mid \mathtt{C} \sqcup \mathtt{C} \mid \leq n\ \mathtt{N_R}.\mathtt{C} \mid \geq n\ \mathtt{N_R}.\mathtt{C}$$

  où $\mathtt{N_C} \in \mathit{NomsConcepts}$, $\mathtt{N_R} \in \mathit{NomsRoles}$, $n \geq 0$
  - $\leq n\ \mathtt{R}.\mathtt{C}$ = 'au plus n $\mathtt{R}$-successeurs sont dans $\mathtt{C}$'
  - $\geq n\ \mathtt{R}.\mathtt{C}$ = 'au moins n $\mathtt{R}$-successeurs sont dans $\mathtt{C}$'
- interprétation :

$$(\leq n\ \mathtt{R}.\mathtt{C})^{\mathcal{I}} = \left\{a \in \Delta^{\mathcal{I}} \ :\ \mathit{Card}(\{b \in \Delta \ :\ (a,b) \in \mathtt{R}^{\mathcal{I}} \ \underline{\text{et } b \in \mathtt{C}^{\mathcal{I}}}\}) \leq n\right\}$$

$$(\geq n\ \mathtt{R}.\mathtt{C})^{\mathcal{I}} = \ldots$$

- ALCQ est au moins aussi expressif que ALCN :
  - $\leq n\ \mathtt{R} \stackrel{\mathrm{def}}{=} \ldots$
  - $\geq n\ \mathtt{R} \stackrel{\mathrm{def}}{=} \ldots$
  - $\exists \mathtt{R}.\mathtt{C} \stackrel{\mathrm{def}}{=} \ldots$
  - $\forall \mathtt{R}.\mathtt{C} \stackrel{\mathrm{def}}{=} \ldots$
- ALCQ est strictement plus expressif que ALCN
  - il y a des restrictions de cardinalité qualifiées qui ne peuvent pas être exprimées dans ALCN

Outline

Rappel : la logique du premier ordre FOL

- langage :
  - variables objet x, y,...
  - prédicats : $P(t_1, \ldots, t_n)$
    - t_i : termes, construits à partir de variables et fonctions
    - un prédicat particulier : $egal(t_1, t_2)$, écrit $t_1 = t_2$
  - formules complexes : construites avec $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, $\exists$, $\forall$

    $(\exists y \forall x P(x, y)) \rightarrow (\forall x \exists y P(x, y))$
- interprétations $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$:
  - domaine $\Delta^{\mathcal{I}}$ (non vide)
  - interprétation d'une variable: $(x)^{\mathcal{I}} \in \Delta^{\mathcal{I}}$
  - interprétation d'un prédicat n-aire : $(P)^{\mathcal{I}} \subseteq (\Delta^{\mathcal{I}})^n$
    - où on a toujours : $egal^{\mathcal{I}} = \{(d, d) \ : \ d \in \Delta^{\mathcal{I}}\}$
- conditions de vérité :
  - $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}}) \Vdash_{FOL} P(t_1, \ldots, t_n)$ ssi $((t_1)^{\mathcal{I}}, \ldots, (t_n)^{\mathcal{I}}) \in (P)^{\mathcal{I}}$
  - $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}}) \Vdash_{FOL} \forall x \varphi$ ssi $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}_x}) \Vdash_{FOL} \varphi$ pour toute variante $\mathcal{I}_x$ de $\mathcal{I}$ en x
  - ...
- formule φ est valide ssi $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}}) \Vdash_{FOL} \varphi$, pour tout $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$

  formule φ est satisfaisable ssi ...

Traduction de ALC en FOL

$$\boxed{\text{concept } \mathtt{C} \longmapsto \text{formule } \Phi(\mathtt{C}, x) \text{ de FOL}}$$

- le résultat de la traduction $\Phi(\mathtt{C}, x)$ a une seule variable libre : x ("l'individu actuel")
- définition récursive :

$$\begin{aligned}
\Phi(\mathtt{A}, x) &= \mathtt{A}(x) \\
\Phi(\top, x) &= \top \\
\Phi(\bot, x) &= \bot \\
\Phi(\neg\mathtt{C}, x) &= \neg\Phi(\mathtt{C}, x) \\
\Phi(\mathtt{C} \sqcap \mathtt{D}, x) &= \Phi(\mathtt{C}, x) \wedge \Phi(\mathtt{D}, x) \\
\Phi(\mathtt{C} \sqcup \mathtt{D}, x) &= \Phi(\mathtt{C}, x) \vee \Phi(\mathtt{D}, x) \\
\Phi(\forall \mathtt{R}.\mathtt{C}, x) &= \forall y\, (\mathtt{R}(x, y) \rightarrow \Phi(\mathtt{C}, y)) \\
\Phi(\exists \mathtt{R}.\mathtt{C}, x) &= \ldots
\end{aligned}$$

Traduction de ALC en FOL (suite)

Proposition

Pour tout concept $\mathtt{C}$ *et pour toute interprétation* $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ *:*

$$C^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \;:\; \mathcal{I} \Vdash_{FOL} \Phi(\mathtt{C}, a) \right\}$$

démonstration par induction sur la forme de $\mathtt{C}$; on montre :

$$\boxed{\text{pour tout } a \in \Delta^{\mathcal{I}} \;:\; a \in C^{\mathcal{I}} \text{ ssi } \mathcal{I} \Vdash_{FOL} \Phi(\mathtt{C}, a)}$$

1. si $\mathtt{C}$ est un concept atomique $\mathtt{A}$ alors :

$$\begin{array}{lll} a \in \mathtt{A}^{\mathcal{I}} & \text{ssi} & \mathcal{I} \Vdash_{FOL} \mathtt{A}(a) \\ & \text{ssi} & \mathcal{I} \Vdash_{FOL} \Phi(\mathtt{A}, a) \end{array}$$

2. si $\mathtt{C}$ est de la forme $\top$ alors ...
3. si $\mathtt{C}$ est de la forme $\bot$ alors ...
4. si $\mathtt{C}$ est de la forme $\neg\mathtt{D}$ alors :

$$\begin{array}{llll} a \in (\neg\mathtt{D})^{\mathcal{I}} & \text{ssi} & a \notin \mathtt{D}^{\mathcal{I}} & \\ & \text{ssi} & \mathcal{I} \nVdash_{FOL} \Phi(\mathtt{D}, a) & \text{(par H.I.)} \\ & \text{ssi} & \mathcal{I} \Vdash_{FOL} \Phi(\neg\mathtt{D}, a) & \end{array}$$

5. si ...

Traduction de ALC en FOL (suite)

Proposition

Pour tout concept $\mathtt{C}$ *et pour toute interprétation* $(\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ *:*

$$C^{\mathcal{I}} = \left\{ a \in \Delta^{\mathcal{I}} \; : \; \mathcal{I} \Vdash_{FOL} \Phi(\mathtt{C}, a) \right\}$$

démonstration par induction sur la forme de $\mathtt{C}$; on montre :

$$\boxed{\text{pour tout } a \in \Delta^{\mathcal{I}} \; : \; a \in C^{\mathcal{I}} \text{ ssi } \mathcal{I} \Vdash_{FOL} \Phi(\mathtt{C}, a)}$$

1. si $\mathtt{C}$ est un concept atomique $\mathtt{A}$ alors :
$$\begin{array}{lll} a \in \mathtt{A}^{\mathcal{I}} & \text{ssi} & \mathcal{I} \Vdash_{FOL} \mathtt{A}(a) \\ & \text{ssi} & \mathcal{I} \Vdash_{FOL} \Phi(\mathtt{A}, a) \end{array}$$
2. si $\mathtt{C}$ est de la forme $\top$ alors ...
3. si $\mathtt{C}$ est de la forme $\bot$ alors ...
4. si $\mathtt{C}$ est de la forme $\neg\mathtt{D}$ alors :
$$\begin{array}{llll} a \in (\neg\mathtt{D})^{\mathcal{I}} & \text{ssi} & a \notin \mathtt{D}^{\mathcal{I}} & \\ & \text{ssi} & \mathcal{I} \nVdash_{FOL} \Phi(\mathtt{D}, a) & \text{(par H.I.)} \\ & \text{ssi} & \mathcal{I} \Vdash_{FOL} \Phi(\neg\mathtt{D}, a) & \end{array}$$
5. si ...

Traduction de ALC en FOL : exercices

$$
\begin{aligned}
\Phi(\mathtt{A}, x) &= \mathtt{A}(x) \\
\Phi(\top, x) &= \top \\
\Phi(\bot, x) &= \bot \\
\Phi(\neg\mathtt{C}, x) &= \neg\Phi(\mathtt{C}, x) \\
\Phi(\mathtt{C} \sqcap \mathtt{D}, x) &= \Phi(\mathtt{C}, x) \wedge \Phi(\mathtt{D}, x) \\
\Phi(\mathtt{C} \sqcup \mathtt{D}, x) &= \Phi(\mathtt{C}, x) \vee \Phi(\mathtt{D}, x) \\
\Phi(\forall \mathtt{R}.\mathtt{C}, x) &= \forall y\, (\mathtt{R}(x, y) \rightarrow \Phi(\mathtt{C}, y)) \\
\Phi(\exists \mathtt{R}.\mathtt{C}, x) &= \exists y\, (\mathtt{R}(x, y) \wedge \Phi(\mathtt{C}, y))
\end{aligned}
$$

- traduire en FOL :
  1. $\mathtt{Personne} \sqcap \exists \mathtt{parentDe}.\exists \mathtt{parentDe}.\mathtt{Masculin}$
  2. $(\forall \mathtt{parentDe}.\mathtt{Masculin}) \sqcap (\forall \mathtt{parentDe}.\neg\mathtt{Masculin})$
  3. montrer que la traduction est satisfaisable : donner une interprétation de FOL $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ où elle est vraie

Traduction de ALCN en FOL : exercices

- compléter la traduction de ALCN en FOL

$$\Phi(\geq 2\ \mathrm{R}, x) = \ldots$$
$$\Phi(\geq n\ \mathrm{R}, x) = \ldots$$
$$\Phi(\leq n\ \mathrm{R}, x) = \ldots$$

- traduire en FOL
  1. $\mathtt{Personne} \sqcap \leq 2\ \mathtt{parentDe}$

Traduction de ALCN en FOL : solution

$$\Phi(\leq 2\ \mathrm{R}, x) = \forall y_1 \forall y_2 \forall y_3 \Big((\mathrm{R}(x, y_1) \wedge \mathrm{R}(x, y_2) \wedge \mathrm{R}(x, y_3) \rightarrow (y_1 = y_2 \vee y_1 = y_3 \vee y_2 = y_3) \Big)$$

$$\Phi(\leq n\ \mathrm{R}, x) = \forall y_1 \ldots \forall y_{n+1} \left(\left(\mathrm{R}(x, y_1) \wedge \ldots \wedge \mathrm{R}(x, y_{n+1}) \right) \rightarrow \left(\bigvee_{i<j\leq n+1} y_i = y_j \right) \right)$$

$$\Phi(\geq n\ \mathrm{R}, x) = \exists y_1 \ldots \exists y_n \left(\mathrm{R}(x, y_1) \wedge \ldots \wedge \mathrm{R}(x, y_n) \wedge \left(\bigwedge_{i<j\leq n} \neg y_i = y_j \right) \right)$$

Traduction de ALC en logique modale

- logique multimodale K_n :

$$
\begin{aligned}
\Box_{\mathtt{R}}\varphi &= \text{“}\varphi \text{ est vrai dans } \textit{tous} \text{ les états accessibles via } \mathtt{R}\text{”}\\
&= \text{“}\varphi \text{ est nécessaire”}\\
&= \ldots\\
\Diamond_{\mathtt{R}}\varphi &= \text{“}\varphi \text{ est vrai dans } \textit{quelques} \text{ les états accessibles via } \mathtt{R}\text{”}\\
&= \text{“}\varphi \text{ est possible”}\\
&= \ldots
\end{aligned}
$$

- fonction de traduction :

$$
\begin{aligned}
\mu(\mathtt{A}) &= \mathtt{A} && \text{si } \mathtt{A} \text{ est un concept atomique}\\
\mu(\top) &= \top\\
\mu(\bot) &= \bot\\
\mu(\neg\mathtt{C}) &= \neg\mu(\mathtt{C})\\
\mu(\mathtt{C} \sqcap \mathtt{D}) &= \mu(\mathtt{C}) \wedge \mu(\mathtt{D})\\
\mu(\mathtt{C} \sqcup \mathtt{D}) &= \mu(\mathtt{C}) \vee \mu(\mathtt{D})\\
\mu(\forall \mathrm{R}.\mathtt{C}) &= \Box_{\mathrm{R}}\mu(\mathtt{C})\\
\mu(\exists \mathrm{R}.\mathtt{C}) &= \Diamond_{\mathrm{R}}\mu(\mathtt{C})
\end{aligned}
$$

Outline

TBox générale

- $\mathtt{C} \sqsubseteq \mathtt{D}$: 'axiome général d'inclusion de concept'

$$\mathcal{I} \Vdash \mathtt{C} \sqsubseteq \mathtt{D} \text{ ssi } \mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$$

- $\mathtt{C} \equiv \mathtt{D}$: 'axiome général d'équivalence de concept'

$$\mathcal{I} \Vdash \mathtt{C} \equiv \mathtt{D} \text{ ssi } \mathtt{C}^{\mathcal{I}} = \mathtt{D}^{\mathcal{I}}$$

- TBox $\mathcal{T}$ = ensemble fini d'axiomes généraux
- sémantique :

$$\mathcal{I} \Vdash \mathcal{T} \text{ ssi } \mathcal{I} \Vdash t \text{ pour tout } t \in \mathcal{T}$$

- exercices :
  1. exprimer par un axiome que le 2nd argument de la relation $\mathtt{R}$ a la propriété $\mathtt{C}$ ('restriction du codomaine')
     - en FOL: $\forall x \forall y (R(x, y) \to \mathtt{C}(y))$
  2. exprimer par un axiome que le 1er argument de la relation $\mathtt{R}$ a la propriété $\mathtt{C}$ ('restriction du domaine')
     - en FOL : $\forall x \forall y (R(x, y) \to \mathtt{C}(x))$

TBox générale

- $\mathtt{C} \sqsubseteq \mathtt{D}$: ‘axiome général d’inclusion de concept’

$$\mathcal{I} \Vdash \mathtt{C} \sqsubseteq \mathtt{D} \text{ ssi } \mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$$

- $\mathtt{C} \equiv \mathtt{D}$: ‘axiome général d’équivalence de concept’

$$\mathcal{I} \Vdash \mathtt{C} \equiv \mathtt{D} \text{ ssi } \mathtt{C}^{\mathcal{I}} = \mathtt{D}^{\mathcal{I}}$$

- TBox $\mathcal{T}$ = ensemble fini d’axiomes généraux
- sémantique :

$$\mathcal{I} \Vdash \mathcal{T} \text{ ssi } \mathcal{I} \Vdash t \text{ pour tout } t \in \mathcal{T}$$

- exercices :
  1. exprimer par un axiome que le 2nd argument de la relation $\mathtt{R}$ a la propriété $\mathtt{C}$ (‘restriction du codomaine’)
     - en FOL: $\forall x \forall y (R(x, y) \to \mathtt{C}(y))$
  2. exprimer par un axiome que le 1er argument de la relation $\mathtt{R}$ a la propriété $\mathtt{C}$ (‘restriction du domaine’)
     - en FOL : $\forall x \forall y (R(x, y) \to \mathtt{C}(x))$

TBox acyclique

- les axiomes d'inclusion peuvent être éliminés :
  - remplacer $\mathtt{C} \sqsubseteq \mathtt{D}$ par $\mathtt{C} \equiv \mathtt{D} \sqcap \mathtt{D}'$, où D' est nouveau
  - équivalent à la TBox d'origine $\mathcal{T}$ *en ce qui concerne le langage de* $\mathcal{T}$
- $\mathtt{C} \equiv \mathtt{D}$ est une définition ssi C est atomique
  - $\mathtt{etreOncle} \equiv \mathtt{Masculin} \sqcap \exists \mathtt{frereDe}.\exists \mathtt{parentDe}.\top$

$\Rightarrow$ problème : $\mathtt{A} \equiv \mathtt{A} \sqcup \mathtt{C}$ n'est pas vraiment une définition...

- une TBox générale $\mathcal{T}$ est une TBox acyclique ssi
  1. $\mathcal{T}$ ne contient que des définitions
  2. pas de définitions multiples
     - interdit : $\mathtt{A} \equiv \mathtt{C}, \mathtt{A} \equiv \mathtt{C}' \in \mathcal{T}$ et $\mathtt{C} \neq \mathtt{C}'$
  3. pas de cycles
     - interdit : $\mathtt{A}_1 \equiv \ldots \mathtt{A}_2 \ldots, \mathtt{A}_2 \equiv \ldots \mathtt{A}_3 \ldots, \ldots, \mathtt{A}_n \equiv \ldots \mathtt{A}_1 \ldots$
- si $\mathtt{A} \equiv \mathtt{C} \in \mathcal{T}$ alors A est un concept non-primitif

$\Rightarrow$ fréquent en pratique (ex. : SNOMED CT)

$\Rightarrow$ intéressant pour le calcul

Outline

ABox = *assertion box*

- pour un ensemble donné de noms d'individus $\mathit{NomsIndividus} = \{a, b, \ldots\}$
  - exemple : $\mathit{NomsIndividus} = \{\mathtt{Alice}, \mathtt{Bob}, \mathtt{Charles}, \ldots\}$
- $\mathtt{C}(a)$: instance de concept
  - $\mathtt{Masculin}(\mathtt{Charles})$
  - $(\mathtt{Masculin} \sqcup \mathtt{Feminine})(\mathtt{Dominique})$
  - $(\mathtt{Masculin} \sqcap \mathtt{Personne})(\mathtt{Charles})$
- $\mathtt{R}(a, b)$: instance de rôle
  - $\mathtt{parentDe}(\mathtt{Alice}, \mathtt{Charles})$
- ABox $\mathcal{A}$ = ensemble fini de $\mathtt{C}(a)$ et $\mathtt{R}(a, b)$
- sémantique : extension de la fonction d'interprétation $\mathcal{I}$ aux aux noms d'individu
  - $(.)^{\mathcal{I}}$: $\mathit{NomsIndividus} \longrightarrow \Delta^{\mathcal{I}}$
    - $\mathtt{Alice}^{\mathcal{I}} \in \Delta^{\mathcal{I}}$, $\mathtt{Bob}^{\mathcal{I}} \in \Delta^{\mathcal{I}}, \ldots$
    - hypothèse de nom unique : si $a \neq b$ alors $a^{\mathcal{I}} \neq b^{\mathcal{I}}$
      exemple : $\mathtt{Bob}^{\mathcal{I}} \neq \mathtt{Charles}^{\mathcal{I}}$
  - $\mathcal{I} \Vdash \mathcal{A}$ ssi
    1. pour tout $\mathtt{C}(a) \in \mathcal{A}$: $a^{\mathcal{I}} \in \mathtt{C}^{\mathcal{I}}$
    2. pour tout $\mathtt{R}(a, b) \in \mathcal{A}$: $(a^{\mathcal{I}}, b^{\mathcal{I}}) \in \mathtt{R}^{\mathcal{I}}$

Exemple d'une KB généalogique

$$\mathcal{T}_{gen} = \{ \begin{array}{rcl} \texttt{Femme} & \equiv & \texttt{Personne} \sqcap \texttt{Feminine}, \\ \texttt{Homme} & \equiv & \texttt{Personne} \sqcap \texttt{Masculin}, \\ \texttt{Mere} & \equiv & \texttt{Femme} \sqcap \exists \texttt{parentDe.Personne}, \\ \texttt{Pere} & \equiv & \texttt{Homme} \sqcap \exists \texttt{parentDe.Personne}, \\ \texttt{Parent} & \equiv & \texttt{Mere} \sqcup \texttt{Pere}, \\ \texttt{MereSansFille} & \equiv & \texttt{Mere} \sqcap \forall \texttt{parentDe.} \neg \texttt{Femme} \end{array} \}$$

$$\mathcal{A}_{gen} = \{ \begin{array}{l} \texttt{Femme(Alice)}, \\ \texttt{Homme(Bob)}, \\ \texttt{parentDe(Alice, Charles)}, \\ \texttt{parentDe(Alice, Denis)}, \\ \texttt{parentDe(Bob, Charles)} \end{array} \}$$

Différence avec les bases de donnés

- DL : hypothèse du monde ouvert (OWA)
  $\Rightarrow$ Alice peut avoir d'autres enfants
  - BD : hypothèse du monde clos (CWA)
    $\Rightarrow$ Charles et Denis sont les seuls enfants d'Alice
- DL : pas toujours hypothèse du nom unique (UNA)
  $\Rightarrow$ Charles et Denis peuvent désigner la même personne
  - BD : toujours hypothèse du nom unique
    $\Rightarrow$ Alice a au moins deux enfants
    $\Rightarrow$ avec la CWA : Alice a exactement deux enfants

Outline

Tâches de raisonnement sur les concepts

C est satisfaisable ssi il existe une interprétation $\mathcal{I}$ tel que $C^{\mathcal{I}} \neq \emptyset$

- exemples :
  - $\mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\bot$ est satisfaisable
  - $\forall \mathtt{parentDe}.C \sqcap \exists \mathtt{parentDe}.\neg C$ est insatisfaisable
  - $\forall \mathtt{parentDe}.C \sqcap \forall \mathtt{parentDe}.\neg C$ est satisfaisable (!)
- subsomption de concepts :
  - $C \sqsubseteq D$ ssi $C^{\mathcal{I}} \subseteq D^{\mathcal{I}}$ pour tout interprétation $\mathcal{I}$
  - réductible à un test de satisfaisabilité
    - $C \sqsubseteq D$ ssi $C \sqcap \neg D$ est insatisfaisable
- équivalence :
  - C et D sont équivalents ssi $C^{\mathcal{I}} = D^{\mathcal{I}}$ pour tout interprétation $\mathcal{I}$
  - réductible à deux tests de subsomption
- exclusivité (*disjointness*) :
  - C et D sont disjoints ssi $C^{\mathcal{I}} \cap D^{\mathcal{I}} = \emptyset$ pour tout interprétation $\mathcal{I}$
  - réductible à un test de satisfaisabilité
    - C et D sont disjoints ssi $C \sqcap D$ est insatisfaisable

Tâches de raisonnement sur les concepts

C est satisfaisable ssi il existe une interprétation $\mathcal{I}$ tel que $\mathtt{C}^{\mathcal{I}} \neq \emptyset$

- exemples :
  - $\mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\bot$ est satisfaisable
  - $\forall \mathtt{parentDe}.\mathtt{C} \sqcap \exists \mathtt{parentDe}.\neg \mathtt{C}$ est insatisfaisable
  - $\forall \mathtt{parentDe}.\mathtt{C} \sqcap \forall \mathtt{parentDe}.\neg \mathtt{C}$ est satisfaisable (!)
- subsomption de concepts :
  - $\mathtt{C} \sqsubseteq \mathtt{D}$ ssi $\mathtt{C}^{\mathcal{I}} \subseteq \mathtt{D}^{\mathcal{I}}$ pour tout interprétation $\mathcal{I}$
  - réductible à un test de satisfaisabilité
    - $\mathtt{C} \sqsubseteq \mathtt{D}$ ssi $\mathtt{C} \sqcap \neg \mathtt{D}$ est insatisfaisable
- équivalence :
  - C et D sont équivalents ssi $\mathtt{C}^{\mathcal{I}} = \mathtt{D}^{\mathcal{I}}$ pour tout interprétation $\mathcal{I}$
  - réductible à deux tests de subsomption
- exclusivité (*disjointness*) :
  - C et D sont disjoints ssi $\mathtt{C}^{\mathcal{I}} \cap \mathtt{D}^{\mathcal{I}} = \emptyset$ pour tout interprétation $\mathcal{I}$
  - réductible à un test de satisfaisabilité
    - C et D sont disjoints ssi $\mathtt{C} \sqcap \mathtt{D}$ est insatisfaisable

Tâches de raisonnement p.r.à une TBox

C est satisfaisable p.r.à $\mathcal{T}$ ssi il existe $\mathcal{I}$ tel que $\mathcal{I} \Vdash \mathcal{T}$ et $C^{\mathcal{I}} \neq \emptyset$

- exemple :
  - $\texttt{Mere} \sqcap \forall \texttt{parentDe}.\bot$ est insatisfaisable p.r.à $\mathcal{T}_{gen}$
- subsomption p.r.à $\mathcal{T}$ ...
- équivalence p.r.à $\mathcal{T}$ ...
- exclusivité (*disjointness*) p.r.à $\mathcal{T}$ ...
- classification = calculer toute hiérarchie de subsomption d'une TBox (graphe)

$\Rightarrow$ tout se réduit à des tests de satisfaisabilité p.r.à $\mathcal{T}$

$\Rightarrow$ et si la TBox $\mathcal{T}$ est acyclique alors on peut faire encore plus simple ...

Tâches de raisonnement p.r.à une TBox

$\boxed{\texttt{C} \text{ est satisfaisable p.r.à } \mathcal{T} \text{ ssi il existe } \mathcal{I} \text{ tel que } \mathcal{I} \Vdash \mathcal{T} \text{ et } \texttt{C}^{\mathcal{I}} \neq \emptyset}$

- exemple :
  - $\texttt{Mere} \sqcap \forall\texttt{parentDe}.\bot$ est insatisfaisable p.r.à $\mathcal{T}_{gen}$
- subsomption p.r.à $\mathcal{T}$ ...
- équivalence p.r.à $\mathcal{T}$ ...
- exclusivité (*disjointness*) p.r.à $\mathcal{T}$ ...
- classification = calculer toute hiérarchie de subsomption d'une TBox (graphe)

$\Rightarrow$ tout se réduit à des tests de satisfaisabilité p.r.à $\mathcal{T}$

$\Rightarrow$ et si la TBox $\mathcal{T}$ est acyclique alors on peut faire encore plus simple ...

Comment se débarasser d'une TBox acyclique

- $C^{\mathcal{T}}$ = expansion de C par $\mathcal{T}$:
  1. remplacer chaque concept non-primitif A_i dans C par la définition de A_i dans $\mathcal{T}$
  2. itérer jusqu'à ce qu'il n'y a plus de concepts non-primitifs
     $\Rightarrow$ termine car $\mathcal{T}$ est acyclique (*point fixe*)
- $\mathcal{A}^{\mathcal{T}} = \{C^{\mathcal{T}}(a) \ : \ C(a) \in \mathcal{A}\} \cup \{R(a,b) \ : \ R(a,b) \in \mathcal{A}\}$
- exercice :

$$\mathcal{T}_{gen} = \{ \begin{array}{rcl} \mathtt{Femme} & \equiv & \mathtt{Personne} \sqcap \mathtt{Feminine}, \\ \mathtt{Homme} & \equiv & \mathtt{Personne} \sqcap \mathtt{Masculin}, \\ \mathtt{Mere} & \equiv & \mathtt{Femme} \sqcap \exists \mathtt{parentDe.Personne}, \\ \mathtt{Pere} & \equiv & \mathtt{Homme} \sqcap \exists \mathtt{parentDe.Personne}, \\ \mathtt{Parent} & \equiv & \mathtt{Mere} \sqcup \mathtt{Pere}, \\ \mathtt{MereSansFille} & \equiv & \mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\neg \mathtt{Femme} \end{array} \}$$

  1. identifier les concepts primitifs
  2. trouver les expansions de tous les concepts non-primitifs
  3. trouver l'expansion de $\mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\bot$

Comment se débarasser d'une TBox acyclique

- $\mathtt{C}^{\mathcal{T}}$ = expansion de $\mathtt{C}$ par $\mathcal{T}$:
  1. remplacer chaque concept non-primitif $\mathtt{A}_i$ dans $\mathtt{C}$ par la définition de $\mathtt{A}_i$ dans $\mathcal{T}$
  2. itérer jusqu'à ce qu'il n'y a plus de concepts non-primitifs
     $\Rightarrow$ termine car $\mathcal{T}$ est acyclique (*point fixe*)
- $\mathcal{A}^{\mathcal{T}} = \{\mathtt{C}^{\mathcal{T}}(\mathtt{a}) \ : \ \mathtt{C}(\mathtt{a}) \in \mathcal{A}\} \cup \{\mathtt{R}(\mathtt{a},\mathtt{b}) \ : \ \mathtt{R}(\mathtt{a},\mathtt{b}) \in \mathcal{A}\}$
- exercice :

$$\mathcal{T}_{gen} = \{ \begin{array}{rcl} \mathtt{Femme} & \equiv & \mathtt{Personne} \sqcap \mathtt{Feminine}, \\ \mathtt{Homme} & \equiv & \mathtt{Personne} \sqcap \mathtt{Masculin}, \\ \mathtt{Mere} & \equiv & \mathtt{Femme} \sqcap \exists \mathtt{parentDe}.\mathtt{Personne}, \\ \mathtt{Pere} & \equiv & \mathtt{Homme} \sqcap \exists \mathtt{parentDe}.\mathtt{Personne}, \\ \mathtt{Parent} & \equiv & \mathtt{Mere} \sqcup \mathtt{Pere}, \\ \mathtt{MereSansFille} & \equiv & \mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\neg\mathtt{Femme} \end{array} \}$$

  1. identifier les concepts primitifs
  2. trouver les expansions de tous les concepts non-primitifs
  3. trouver l'expansion de $\mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\bot$

Comment se débarasser d'une TBox acyclique (suite)

Proposition

Si $\mathcal{T}$ est acyclique alors
$\mathtt{C}$ est satisfaisable p.r.à $\mathcal{T}$ ssi
$\mathtt{C}^{\mathcal{T}}$ est satisfaisable.

- exemple : $\mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\bot$ satisfaisable p.r.à $\mathcal{T}_{gen}$ ssi
  $(\mathtt{Mere} \sqcap \forall \mathtt{parentDe}.\bot)^{\mathcal{T}_{gen}} =$
  $\mathtt{Personne} \sqcap \mathtt{Feminine} \sqcap \exists \mathtt{parentDe}.\mathtt{Personne} \sqcap \forall \mathtt{parentDe}.\bot$
  est satisfaisable tout court
  . . . ce qui n'est pas le cas
- exercice :
  - trouver $\mathcal{T}$ et $\mathtt{A}$ t.q. $\mathtt{A}^{\mathcal{T}}$ est exponentiellement plus long que $\mathcal{T}$

Tâches de raisonnement p.r.à une ABox

une ABox $\mathcal{A}$ est satisfaisable ssi il existe $\mathcal{I}$ tel que $\mathcal{I} \Vdash \mathcal{A}$

- satisfaisabilité p.r.à une TBox :
  - $\mathcal{A}$ satisfaisable p.r.à $\mathcal{T}$ ssi il existe $\mathcal{I}$ tel que $\mathcal{I} \Vdash \mathcal{T}$ et $\mathcal{I} \Vdash \mathcal{A}$
  - si $\mathcal{T}$ est acyclique : $\mathcal{A}$ satisfaisable p.r.à $\mathcal{T}$ ssi $\mathcal{A}^{\mathcal{T}}$ satisfaisable
  - subsume la satisfaisabilité d'un concept
    - $\mathtt{C}$ est satisfaisable ssi $\{\mathtt{C}(\mathtt{a})\}$ est satisfaisable, pour un individu a quelconque
- inférence de propriétés :
  - $\mathcal{T} \cup \mathcal{A} \models \mathtt{C}(\mathtt{a})$ ssi pour tout $\mathcal{I}$, si $\mathcal{I} \Vdash \mathcal{T}$ et $\mathcal{I} \Vdash \mathcal{A}$ alors $\mathcal{I} \Vdash \mathtt{C}(\mathtt{a})$
  - exemple :
    - $\mathcal{T}_{gen} \cup \{\mathtt{GrandMere}(\mathtt{Alice})\} \models \mathtt{Mere}(\mathtt{Alice})$
  - $\mathcal{T} \cup \mathcal{A} \models \mathtt{C}(\mathtt{a})$ ssi $\mathcal{A} \cup \{\neg\mathtt{C}(\mathtt{a})\}$ insatisfaisable p.r.à $\mathcal{T}$
- requête :
  - trouver tous les a tel que $\mathcal{T} \cup \mathcal{A} \models \mathtt{C}(a)$
- réalisation :
  - calculer les noms de concept les plus spécifiques de la TBox pour un individu donné (cf. la classification)
    - Bob est instance de `Personne`, `Masculin`, `Pere`

Tâches de raisonnement p.r.à une ABox

une ABox $\mathcal{A}$ est satisfaisable ssi il existe $\mathcal{I}$ tel que $\mathcal{I} \Vdash \mathcal{A}$

- satisfaisabilité p.r.à une TBox :
  - $\mathcal{A}$ satisfaisable p.r.à $\mathcal{T}$ ssi il existe $\mathcal{I}$ tel que $\mathcal{I} \Vdash \mathcal{T}$ et $\mathcal{I} \Vdash \mathcal{A}$
  - si $\mathcal{T}$ est acyclique : $\mathcal{A}$ satisfaisable p.r.à $\mathcal{T}$ ssi $\mathcal{A}^{\mathcal{T}}$ satisfaisable
  - subsume la satisfaisabilité d'un concept
    - $\mathtt{C}$ est satisfaisable ssi $\{\mathtt{C}(\mathtt{a})\}$ est satisfaisable, pour un individu *a* quelconque
- inférence de propriétés :
  - $\mathcal{T} \cup \mathcal{A} \models \mathtt{C}(\mathtt{a})$ ssi pour tout $\mathcal{I}$, si $\mathcal{I} \Vdash \mathcal{T}$ et $\mathcal{I} \Vdash \mathcal{A}$ alors $\mathcal{I} \Vdash \mathtt{C}(\mathtt{a})$
  - exemple :
    - $\mathcal{T}_{gen} \cup \{\mathtt{GrandMere}(\mathtt{Alice})\} \models \mathtt{Mere}(\mathtt{Alice})$
  - $\mathcal{T} \cup \mathcal{A} \models \mathtt{C}(\mathtt{a})$ ssi $\mathcal{A} \cup \{\neg\mathtt{C}(\mathtt{a})\}$ insatisfaisable p.r.à $\mathcal{T}$
- requête :
  - trouver tous les *a* tel que $\mathcal{T} \cup \mathcal{A} \models \mathtt{C}(a)$
- réalisation :
  - calculer les noms de concept les plus spécifiques de la TBox pour un individu donné (cf. la classification)
    - `Bob` est instance de `Personne`, `Masculin`, `Pere`

Outline

Le langage SROIQ

- essentiellement OWL DL
- ALCQ plus :
  - rôle universel $\mathtt{U}$
  - hierarchies de rôle

    $\mathtt{frereDe} \circ \mathtt{parentDe} \sqsubseteq \mathtt{oncleDe}$
  - nominaux = concepts particuliers désignant un seul individu

    $\exists\mathtt{numeroVol}.\{\mathtt{AF3021}\}$

    $\{\mathtt{Alice}\} \approx \{\mathtt{Anne}\}$ …donc pas de UNA !

    $\{\mathtt{Alice}\} \not\approx \{\mathtt{Anne}\}$ équivalent : $\{\mathtt{Alice}\} \sqcap \{\mathtt{Anne}\} \sqsubseteq \bot$

    $\{\mathtt{Alice}\} \sqcup \{\mathtt{Anne}\} \sqsubseteq \mathtt{Mere}$

    équivalent : $\mathtt{Mere}(\mathtt{Alice}), \mathtt{Mere}(\mathtt{Anne})$

    $\mathtt{Mere} \sqsubseteq \{\mathtt{Alice}\} \sqcup \{\mathtt{Anne}\}$ …ne peut pas être exprimé en ALC !
  - rôles inverses

    $\mathtt{parentDe}^{-}$
  - *Self*

    $\neg\exists\mathtt{parentDe}.\mathit{Self}$

Le langage SROIQ

- grammaire :

$$
\begin{aligned}
\text{rôles} : \mathtt{R} &::= \mathtt{U} \mid \mathtt{N_R} \mid \mathtt{N_R^-} \\
\text{concepts} : \mathtt{C} &::= \mathtt{N_C} \mid \{\mathtt{N_I}\} \mid \top \mid \bot \mid \neg\mathtt{C} \mid \mathtt{C} \sqcap \mathtt{C} \mid \mathtt{C} \sqcup \mathtt{C} \mid \\
&\quad\ \forall\mathtt{R.C} \mid \exists\mathtt{R.C} \mid \leq n\ \mathtt{R.C} \mid \geq n\ \mathtt{R.C} \mid \exists\mathtt{R}.\mathit{Self} \\
\text{assertions} : \alpha &::= \mathtt{C}(\mathtt{N_I}) \mid \mathtt{R}(\mathtt{N_I}, \mathtt{N_I}) \mid \mathtt{N_I} \approx \mathtt{N_I} \mid \mathtt{N_I} \not\approx \mathtt{N_I} \\
\text{axiomes} : \tau &::= \mathtt{C} \sqsubseteq \mathtt{C} \mid \mathtt{C} \equiv \mathtt{C} \mid \\
&\quad\ \mathtt{R} \sqsubseteq \mathtt{R} \mid \mathtt{R} \equiv \mathtt{R} \mid \mathtt{R} \circ \mathtt{R} \sqsubseteq \mathtt{R} \mid \mathit{Disjoint}(\mathtt{R}, \mathtt{R})
\end{aligned}
$$

- en plus :
  - dans $\leq n\ \mathtt{R.C}$, $\geq n\ \mathtt{R.C}$, $\exists\mathtt{R}.\mathit{Self}$, $\mathit{Disjoint}(\mathtt{R}, \mathtt{R})$ les $\mathtt{R}$ doivent être *simples* . . .
  - les inclusions de rôle $\mathtt{R} \sqsubseteq \mathtt{R}$ ne doivent pas générer des cycles (ou seulement des formes simples de dépendances cycliques)

Outline

Décidabilité et complexité

- le problème de satisfaisabilité d'un concept est décidable :
  - PSPACE complet si la TBox est acyclique
  - EXPTIME complet en général

$\Rightarrow$ suite : procédure de décision via tableaux pour la satisfaisabilité d'une ABox

$\Rightarrow$ subsume toutes les autres tâches de raisonnement dans le cas d'une TBox acyclique

La méthode des tableaux

- entrée : ABox $\mathcal{A}$
- sortie : ‘oui’ si $\mathcal{A}$ est satisfaisable, ‘non’ sinon
- hypothèse : $\mathcal{A}$ en forme normale négative
  - négations seulement devant les atomes
- notions et notations :
  - configuration du tableau = ensemble fini d’ABox’es S
    - initialisation : $S = \{\mathcal{A}\}$
  - $S, \mathcal{A}$ à la place de $S \cup \{\mathcal{A}\}$
  - $\mathcal{A}[X]$ = “X est un sous-ensemble de $\mathcal{A}$”
  - format des règles :

$$\frac{S,\ \mathcal{A}[X]}{S,\ \mathcal{A}[Y_1], \ldots, \mathcal{A}[Y_n]}$$

    - si $X \subseteq \mathcal{A}$ alors ajouter Y_i à $\mathcal{A}$
    - applicable si $Y_1 \nsubseteq \mathcal{A}, \ldots$ et $Y_n \nsubseteq \mathcal{A}$

Règles de tableau pour ALC

- règle pour $\sqcap$:

$$\frac{S,\ \mathcal{A}[(\mathtt{C} \sqcap \mathtt{D})(\mathtt{a})]}{S,\ \mathcal{A}[\mathtt{C}(\mathtt{a}), \mathtt{D}(\mathtt{a})]}$$

- règle pour $\sqcup$:

$$\frac{S,\ \mathcal{A}[(\mathtt{C} \sqcup \mathtt{D})(\mathtt{a})]}{S,\ \mathcal{A}[\mathtt{C}(\mathtt{a})],\ \mathcal{A}[\mathtt{D}(\mathtt{a})]}$$

- règle pour $\exists$:

$$\frac{S,\ \mathcal{A}[(\exists \mathtt{R}.\mathtt{C})(\mathtt{a})]}{S,\ \mathcal{A}[\mathtt{R}(\mathtt{a},\mathtt{b}), \mathtt{C}(\mathtt{b})]} \quad ((\mathtt{b}) \notin \mathcal{A})$$

b = nouveau individu

- règle pour $\forall$:

$$\frac{S,\ \mathcal{A}[(\forall \mathtt{R}.\mathtt{C})(\mathtt{a}), \mathtt{R}(\mathtt{a},\mathtt{b})]}{S,\ \mathcal{A}[\mathtt{C}(\mathtt{b})]}$$

Règles de tableau pour ALC

- règle pour $\sqcap$:

$$\frac{S,\ \mathcal{A}[(\mathtt{C} \sqcap \mathtt{D})(\mathtt{a})]}{S,\ \mathcal{A}[\mathtt{C}(\mathtt{a}), \mathtt{D}(\mathtt{a})]}$$

- règle pour $\sqcup$:

$$\frac{S,\ \mathcal{A}[(\mathtt{C} \sqcup \mathtt{D})(\mathtt{a})]}{S,\ \mathcal{A}[\mathtt{C}(\mathtt{a})],\ \mathcal{A}[\mathtt{D}(\mathtt{a})]}$$

- règle pour $\exists$:

$$\frac{S,\ \mathcal{A}[(\exists \mathtt{R}.\mathtt{C})(\mathtt{a})]}{S,\ \mathcal{A}[\mathtt{R}(\mathtt{a}, \mathtt{b}), \mathtt{C}(\mathtt{b})]} \quad ((\mathtt{b}) \notin \mathcal{A})$$

b = nouveau individu

- règle pour $\forall$:

$$\frac{S,\ \mathcal{A}[(\forall \mathtt{R}.\mathtt{C})(\mathtt{a}), \mathtt{R}(\mathtt{a}, \mathtt{b})]}{S,\ \mathcal{A}[\mathtt{C}(\mathtt{b})]}$$

Tableaux : contradictions manifeste

Une ABox $\mathcal{A}$ contient une contradiction manifeste (*clash*) si il existe un a tel que

- $\perp(\mathtt{a}) \in \mathcal{A}$, ou
- $\mathtt{C}(\mathtt{a}) \in \mathcal{A}$ et $\neg\mathtt{C}(\mathtt{a}) \in \mathcal{A}$

Sinon $\mathcal{A}$ est *ouvert*.

- exemple :
  $\mathcal{A} = \{\mathtt{Personne} \sqcap (\exists \mathtt{parentDe.Personne}) \sqcap \forall \mathtt{parentDe.}\perp(\mathtt{a})\}$
  1. déjà en forme normale négative
  2. $\{\{\mathtt{Personne}(\mathtt{a}), \exists\mathtt{parentDe.Personne}(\mathtt{a}), \forall\mathtt{parentDe.}\perp(\mathtt{a})\}\}$
     (règle pour $\sqcap$)
  3. $\{\{\mathtt{Personne}(\mathtt{a}), \exists\mathtt{parentDe.Personne}(\mathtt{a}), \forall\mathtt{parentDe.}\perp(\mathtt{a}), \mathtt{Personne}(\mathtt{b}), \mathtt{parentDe}(\mathtt{a}, \mathtt{b})\}\}$
     (règle pour $\exists$)
  4. $\{\{\mathtt{Personne}(\mathtt{a}), \exists\mathtt{parentDe.Personne}(\mathtt{a}), \forall\mathtt{parentDe.}\perp(\mathtt{a}), \mathtt{Personne}(\mathtt{b}), \mathtt{parentDe}(\mathtt{a}, \mathtt{b}), \perp(\mathtt{b})\}\}$
     (règle pour $\forall$)
  5. contradiction manifeste : $\perp(\mathtt{b})$

Tableaux : contradictions manifeste

Une ABox $\mathcal{A}$ contient une contradiction manifeste (*clash*) si il existe un a tel que

- $\bot(\mathtt{a}) \in \mathcal{A}$, ou
- $\mathtt{C}(\mathtt{a}) \in \mathcal{A}$ et $\neg\mathtt{C}(\mathtt{a}) \in \mathcal{A}$

Sinon $\mathcal{A}$ est *ouvert*.

- exemple :
  $\mathcal{A} = \{\mathtt{Personne} \sqcap (\exists\mathtt{parentDe.Personne}) \sqcap \forall\mathtt{parentDe}.\bot(\mathtt{a})\}$
  1. déjà en forme normale négative
  2. $\{\{\mathtt{Personne}(\mathtt{a}), \exists\mathtt{parentDe.Personne}(\mathtt{a}), \forall\mathtt{parentDe}.\bot(\mathtt{a})\}\}$
     (règle pour $\sqcap$)
  3. $\{\{\mathtt{Personne}(\mathtt{a}), \exists\mathtt{parentDe.Personne}(\mathtt{a}), \forall\mathtt{parentDe}.\bot(\mathtt{a}), \mathtt{Personne}(\mathtt{b}), \mathtt{parentDe}(\mathtt{a}, \mathtt{b})\}\}$
     (règle pour $\exists$)
  4. $\{\{\mathtt{Personne}(\mathtt{a}), \exists\mathtt{parentDe.Personne}(\mathtt{a}), \forall\mathtt{parentDe}.\bot(\mathtt{a}), \mathtt{Personne}(\mathtt{b}), \mathtt{parentDe}(\mathtt{a}, \mathtt{b}), \bot(\mathtt{b})\}\}$
     (règle pour $\forall$)
  5. contradiction manifeste : $\bot(\mathtt{b})$

Raisonnement : exercice complet (1)

$$\boxed{\{\mathtt{A} \sqsubseteq \exists\mathtt{R}.\mathtt{B}\} \cup \{\mathtt{A}(\mathtt{a})\} \models^{?} (\exists\mathtt{R}.\mathtt{B})(\mathtt{a})}$$

(tâche = inférence de propriétés)

1. transformation en le problème de satisfaisabilité d'une ABox :

$$\boxed{\{\mathtt{A}(\mathtt{a})\} \cup \{(\neg\exists\mathtt{R}.\mathtt{B})(\mathtt{a})\} \text{ satisfaisable p.r.à } \{\mathtt{A} \sqsubseteq \exists\mathtt{R}.\mathtt{B}\} \text{ ?}}$$

2. éliminer $\sqsubseteq$:

$$\boxed{\{\mathtt{A}(\mathtt{a}), (\neg\exists\mathtt{R}.\mathtt{B})(\mathtt{a})\} \text{ satisfaisable p.r.à } \{\mathtt{A} \equiv \mathtt{A}' \sqcap \exists\mathtt{R}.\mathtt{B}\} \text{ ?}}$$

$\Rightarrow$ TBox acyclique !

3. expansion de la ABox par la TBox : $\mathtt{A}^{\{\mathtt{A}\equiv\mathtt{A}'\sqcap\exists\mathtt{R}.\mathtt{B}\}} = \mathtt{A}' \sqcap \exists\mathtt{R}.\mathtt{B}$

$$\boxed{\{(\mathtt{A}' \sqcap \exists\mathtt{R}.\mathtt{B})(\mathtt{a})\} \cup \{(\neg\exists\mathtt{R}.\mathtt{B})(\mathtt{a})\} \text{ satisfaisable ?}}$$

4. mise en forme normale négative :

$$\boxed{\{(\mathtt{A}' \sqcap \exists\mathtt{R}.\mathtt{B})(\mathtt{a}), (\forall\mathtt{R}.\neg\mathtt{B})(\mathtt{a})\} \text{ satisfaisable ?}}$$

5. construction d'un tableau . . .

Raisonnement : exercice complet (1)

$$\boxed{\{A \sqsubseteq \exists R.B\} \cup \{A(a)\} \models^? (\exists R.B)(a)}$$

(tâche = inférence de propriétés)

1. transformation en le problème de satisfaisabilité d'une ABox :

$$\boxed{\{A(a)\} \cup \{(\neg\exists R.B)(a)\} \text{ satisfaisable p.r.à } \{A \sqsubseteq \exists R.B\} \text{ ?}}$$

2. éliminer $\sqsubseteq$:

$$\boxed{\{A(a), (\neg\exists R.B)(a)\} \text{ satisfaisable p.r.à } \{A \equiv A' \sqcap \exists R.B\} \text{ ?}}$$

$\Rightarrow$ TBox acyclique !

3. expansion de la ABox par la TBox : $A^{\{A \equiv A' \sqcap \exists R.B\}} = A' \sqcap \exists R.B$

$$\boxed{\{(A' \sqcap \exists R.B)(a)\} \cup \{(\neg\exists R.B)(a)\} \text{ satisfaisable ?}}$$

4. mise en forme normale négative :

$$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$$

5. construction d'un tableau . . .

Raisonnement : exercice complet (1)

$$\boxed{\{A \sqsubseteq \exists R.B\} \cup \{A(a)\} \models^{?} (\exists R.B)(a)}$$

(tâche = inférence de propriétés)

1. transformation en le problème de satisfaisabilité d'une ABox :

   $\boxed{\{A(a)\} \cup \{(\neg\exists R.B)(a)\}$ satisfaisable p.r.à $\{A \sqsubseteq \exists R.B\}$?$}$

2. éliminer $\sqsubseteq$:

   $\boxed{\{A(a), (\neg\exists R.B)(a)\}$ satisfaisable p.r.à $\{A \equiv A' \sqcap \exists R.B\}$?$}$

   $\Rightarrow$ TBox acyclique !

3. expansion de la ABox par la TBox : $A^{\{A \equiv A' \sqcap \exists R.B\}} = A' \sqcap \exists R.B$

   $\boxed{\{(A' \sqcap \exists R.B)(a)\} \cup \{(\neg\exists R.B)(a)\}$ satisfaisable ?$}$

4. mise en forme normale négative :

   $\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ satisfaisable ?$}$

5. construction d'un tableau ...

Raisonnement : exercice complet (1)

$$\boxed{\{A \sqsubseteq \exists R.B\} \cup \{A(a)\} \models^{?} (\exists R.B)(a)}$$

(tâche = inférence de propriétés)

1. transformation en le problème de satisfaisabilité d'une ABox :

   $\boxed{\{A(a)\} \cup \{(\neg\exists R.B)(a)\} \text{ satisfaisable p.r.à } \{A \sqsubseteq \exists R.B\}\ ?}$

2. éliminer $\sqsubseteq$:

   $\boxed{\{A(a), (\neg\exists R.B)(a)\} \text{ satisfaisable p.r.à } \{A \equiv A' \sqcap \exists R.B\}\ ?}$

   $\Rightarrow$ TBox acyclique !

3. expansion de la ABox par la TBox : $A^{\{A \equiv A' \sqcap \exists R.B\}} = A' \sqcap \exists R.B$

   $\boxed{\{(A' \sqcap \exists R.B)(a)\} \cup \{(\neg\exists R.B)(a)\} \text{ satisfaisable ?}}$

4. mise en forme normale négative :

   $\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

5. construction d'un tableau . . .

Raisonnement : exercice complet (1)

$$\boxed{\{A \sqsubseteq \exists R.B\} \cup \{A(a)\} \models^{?} (\exists R.B)(a)}$$

(tâche = inférence de propriétés)

1. transformation en le problème de satisfaisabilité d'une ABox :

   $\boxed{\{A(a)\} \cup \{(\neg\exists R.B)(a)\} \text{ satisfaisable p.r.à } \{A \sqsubseteq \exists R.B\} \text{ ?}}$

2. éliminer $\sqsubseteq$:

   $\boxed{\{A(a), (\neg\exists R.B)(a)\} \text{ satisfaisable p.r.à } \{A \equiv A' \sqcap \exists R.B\} \text{ ?}}$

   $\Rightarrow$ TBox acyclique !

3. expansion de la ABox par la TBox : $A^{\{A \equiv A' \sqcap \exists R.B\}} = A' \sqcap \exists R.B$

   $\boxed{\{(A' \sqcap \exists R.B)(a)\} \cup \{(\neg\exists R.B)(a)\} \text{ satisfaisable ?}}$

4. mise en forme normale négative :

   $\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

5. construction d'un tableau ...

Raisonnement : exercice complet (2)

$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b)\}\}$
   (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b), (\neg B)(b)\}\}$
   (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : exercice complet (2)

$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b)\}\}$ (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b), (\neg B)(b)\}\}$ (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : exercice complet (2)

$\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ satisfaisable ?

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b)\}\}$ (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b), (\neg B)(b)\}\}$ (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : exercice complet (2)

$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b)\}\}$ (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b), (\neg B)(b)\}\}$ (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : exercice complet (2)

$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b)\}\}$
   (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b), (\neg B)(b)\}\}$
   (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : exercice complet (2)

$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b)\}\}$
   (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a,b), B(b), (\neg B)(b)\}\}$
   (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : exercice complet (2)

$\boxed{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}\}$
2. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a)\}\}$ (règle pour $\sqcap$)
3. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a, b), B(b)\}\}$ (règle pour $\exists$; b nouveau)
4. $\{\{A'(a), (\exists R.B)(a), (\forall R.\neg B)(a), R(a, b), B(b), (\neg B)(b)\}\}$ (règle pour $\forall$)

$\Rightarrow$ contradiction manifeste

$\Rightarrow$ la ABox $\{(A' \sqcap \exists R.B)(a), (\forall R.\neg B)(a)\}$ est insatisfaisable

$\Rightarrow$ de la KB $\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\}$ on peut inférer $(\exists R.B)(a)$:

$$\{A \equiv A' \sqcap \exists R.B\} \cup \{A(a)\} \models (\exists R.B)(a)$$

Raisonnement : un autre exercice complet (1)

$$\boxed{\{\mathtt{A} \equiv (\exists \mathrm{R}.\mathrm{B}_1) \sqcap (\exists \mathrm{R}.\mathrm{B}_2)\} \models^? \mathtt{A} \sqsubseteq \exists \mathrm{R}.(\mathrm{B}_1 \sqcap \mathrm{B}_2)}$$

(tâche = subsomption de concepts p.r.à une TBox)

1. transformation en problème de satisfaisabilité de concept p.r.à une TBox :

   $\boxed{\mathtt{A} \sqcap \neg\exists \mathrm{R}.(\mathrm{B}_1 \sqcap \mathrm{B}_2) \text{ satisfaisable p.r.à } \{\mathtt{A} \equiv (\exists \mathrm{R}.\mathrm{B}_1) \sqcap (\exists \mathrm{R}.\mathrm{B}_2)\} \text{ ?}}$
2. expansion du concept par la TBox (qui est acyclique) :

   $\boxed{(\exists \mathrm{R}.\mathrm{B}_1) \sqcap (\exists \mathrm{R}.\mathrm{B}_2) \sqcap \neg\exists \mathrm{R}.(\mathrm{B}_1 \sqcap \mathrm{B}_2) \text{ satisfaisable ?}}$
3. transformation en problème de satisfaisabilité d'une ABox :

   $\boxed{\{((\exists \mathrm{R}.\mathrm{B}_1) \sqcap (\exists \mathrm{R}.\mathrm{B}_2) \sqcap \neg\exists \mathrm{R}.(\mathrm{B}_1 \sqcap \mathrm{B}_2))(a)\} \text{ satisfaisable ?}}$
4. mise en forme normale négative :

   $\boxed{\{((\exists \mathrm{R}.\mathrm{B}_1) \sqcap (\exists \mathrm{R}.\mathrm{B}_2) \sqcap \forall \mathrm{R}.(\neg \mathrm{B}_1 \sqcup \neg \mathrm{B}_2))(a)\} \text{ satisfaisable ?}}$
5. construction d'un tableau . . .

Raisonnement : un autre exercice complet (2)

$\boxed{\{((\exists R.B_1) \sqcap (\exists R.B_2) \sqcap \forall R.(\neg B_1 \sqcup \neg B_2))(a)\} \text{ satisfaisable ?}}$

construction du tableau :

1. $\{ \{(\exists R.B_1) \sqcap (\exists R.B_2) \sqcap \forall R.(\neg B_1 \sqcup \neg B_2)(a)\} \}$
2. $\{ \{((\exists R.B_1)(a), (\exists R.B_2)(a), (\forall R.(\neg B_1 \sqcup \neg B_2))(a)\} \}$

   (règle pour $\sqcap$, deux fois)
3. $\{ \{(\exists R.B_1)(a), (\exists R.B_2)(a), (\forall R.(\neg B_1 \sqcup \neg B_2))(a),$
   $R(a, b_1), B_1(b_1), R(a, b_2), B_2(b_2)\} \}$ (règle pour $\exists$, deux fois)
4. $\{ \{(\exists R.B_1)(a), (\exists R.B_2)(a), (\forall R.(\neg B_1 \sqcup \neg B_2))(a),$
   $R(a, b_1), B_1(b_1), R(a, b_2), B_2(b_2),$
   $(\neg B_1 \sqcup \neg B_2)(b_1), (\neg B_1 \sqcup \neg B_2)(b_2)\} \} = \{\mathcal{A}_0\}$

   (règle pour $\forall$, deux fois)
5. $\{ \mathcal{A}_0 \cup \{(\neg B_1)(b_1)\}, \mathcal{A}_0 \cup \{(\neg B_2)(b_1)\} \}$ (règle pour $\sqcup$)
6. $\{ \mathcal{A}_0 \cup \{(\neg B_1)(b_1)\} \cup \{(\neg B_1)(b_2)\}, \mathcal{A}_0 \cup \{(\neg B_1)(b_1)\} \cup \{(\neg B_2)(b_2)\},$
   $\mathcal{A}_0 \cup \{(\neg B_2)(b_1)\} \cup \{(\neg B_1)(b_2)\}, \mathcal{A}_0 \cup \{(\neg B_2)(b_1)\} \cup \{(\neg B_2)(b_2)\} \}$

   (règle pour $\sqcup$, deux fois)

$\Rightarrow$ 3ème ABox ouverte $\Rightarrow$ satisfaisable $\Rightarrow \mathcal{T} \not\models A \sqsubseteq \exists R.(B_1 \sqcap B_2)$

Tableaux pour ALC : la grande finale

- La configuration S est satisfaisable ssi il existe une ABox $\mathcal{A}_i \in S$ et une interprétation $\mathcal{I}$ t.q. $\mathcal{I} \Vdash \mathcal{A}_i$.
- La configuration S est saturée ssi plus aucune règle ne peut être appliquée

Proposition (terminaison)

Pour toute entrée $\mathcal{A}$, la procédure de construction de tableau termine par une configuration saturée.

Proposition (adéquation)

Si la ABox $\mathcal{A}$ est satisfaisable alors toutes les configurations obtenues à partir de $\{\mathcal{A}\}$ sont ouvertes.

Proposition (complétude)

Si S est une configuration saturée et ouverte alors il existe une ABox $\mathcal{A} \in S$ tel que $\mathcal{A}$ est satisfaisable.

Tableaux pour des extensions de ALC

- ALCN
- inclusion de concepts généraux (cycliques)
  - requiert test de boucle
  - EXPTIME complet
- inclusion de rôles
- rôles transitifs
- réstrictions de cardinalité qualifiées
  - indécidable si combiné avec intersection de rôles !
  - indécidable si combiné avec rôles transitifs !
- constructeur 'un-parmi $a_1, \ldots, a_n$' : $\{a_1, \ldots, a_n\}$
  - $(\{a_1, \ldots, a_n\})^{\mathcal{I}} = \{a_1^{\mathcal{I}}, \ldots, a_n^{\mathcal{I}}\}$
  - exemple : $\texttt{Personne} \equiv \{\texttt{Alice}, \texttt{Bob}, \texttt{Charles}\}$
- domaines concrètes
  - nombres entiers : $\texttt{Adulte} \equiv \texttt{Personne} \sqcap \exists \texttt{ageDe}.{\geq}18$
  - ...

Conclusion sur la partie DL

- DL = représenter + raisonner
- DL = plusieurs logiques
  - expressivité (quelles réstrictions du langage permis par le domaine ?)
  - complexité (P, NP, PSPACE, EXPTIME)
- DL base de OWL (OWL DL)
- recherches actuelles :
  - trouver des langages d'interrogation (*query languages*) à complexité basse
    - restrictions drastiques du langage : “DL-lite”
  - comment réviser une TBox ?
  - comment mettre à jour une ABox ? comment réparer une ABox qui est devenue insatisfaisable p.r.à une TBox ?
    - $\{\mathtt{Mere}(\mathtt{Alice}), \forall \mathtt{parentDe}.\bot(\mathtt{Alice})\}$ insatisfaisable p.r.à $\mathcal{T}_{gen}$
  - comment aligner deux TBox ?