

Cours AH

Introduction aux logiques de description et logiques modales

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

(en partie repris du cours de Nicola Olivetti, Univ. Aix-Marseille)

Univ. Toulouse, M2 IAFA
Cours “Représentation des connaissances en logique : logique modale et ontologies”
2023-24

Introduction : resources

- livres :

- Franz Baader, Ian Horrocks, Carsten Lutz, and Ulrike Sattler

An Introduction to Description Logic

Cambridge University Press, 2017

<http://dltextbook.org>

- Brian Chellas *Modal logic: an introduction*

Cambridge University Press, 1995

- *le site web pour les logiques de description :*

<http://dl.kr.org>

⇒ DL workshop

⇒ conférences afférentes

⇒ entrepôt d'ontologies

Logiques de description et logiques modales : késako ?

- ① logiques de description = restriction de la syntaxe de la logique du premier ordre
 - généralisation des bases de données relationnelles (≥ 1980)
 - idée centrale : *restreindre la quantification*
 - dans $\forall x\varphi$ et $\exists x\varphi$, la formule φ doit être d'une certaine forme
 - fragment décidable de la logique du premier ordre
 - syntaxe sans variables (!)
- ② logiques modales = extension de la syntaxe de la logique propositionnelle
 - opérateurs *modaux*
 - diverses modalités :
 - "il est nécessaire que φ " / "il est possible que φ " (~1912)
 - " φ sera vrai toujours" / " φ sera vrai parfois" (~1960)
 - "l'agent i croit que φ " / "il est compatible avec les croyances de i que φ " (~1960)
 - "l'agent i sait que φ " / "... " (~1960)
 - "il est obligatoire que φ " / "il est permis que φ " (~1970)
 - en commun : opérateurs non *vérifonctionnels*
 - la valeur de vérité de "il est nécessaire que φ " n'est pas fonction de la valeur de vérité de φ

Logiques de description et logiques modales : késako ?

en résumé :

- motivations et applications très différentes
- . . . mais formellement très proches
 - même sémantique
 - plus expressifs que la logique propositionnelle
 - moins expressifs que la logique du premier ordre
- il existe des traductions dans les deux sens

Logiques de description : 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

Logiques de description : 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 ...

Logiques de description : 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)

Logiques de description : 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 :

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

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

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

Logiques de description : 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)

Logiques de description : l'aspect raisonnement

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

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

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

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

- 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 subsumption de concepts
 - 4 satisfaisabilité (pas de contradictions)
 - 5 non-vacuité de concepts ; non-redondance de concepts

Logiques de description : avantages

- 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
 - bases de données (modèles entité-relation exprimables en DL)
 - web sémantique
 - 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

Logiques de description : complexité vs. expressivité

- restrictions de l'expressivité :
 - pas de négation '¬' ; pas de disjonction '∨' ; ...
 - pas d'inclusion de rôles '⊆' ; pas de rôles complexes ; ...
 - dans $\exists R.C$ il faut que $C = \top$; ...

- logique expressive \Rightarrow **raisonnement complexe**

PTIME = temps polynomial

NPTIME = temps polynomial non-déterministe

PSPACE = espace polynomial (PSPACE = NSPACE)

EXPTIME = temps exponentiel

NEXPTIME = temps exponentiel non-déterministe

EXPSPACE = espace exponentiel

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

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

Syntaxe

- ALC = **A**ttributive **C**oncept **L**anguage with **C**omplements
[Schmidt-Schauß & Smolka, 1991]

N.B. : plutôt que 'langage', devrait s'appeller *logique*
(logique = langage + sémantique)

- convention :
 - A, A_j , ... : concepts atomiques
 - dans les exemples : commence par majuscule (Parent, ...)
 - C, D, C_j , ... : concepts arbitraires
 - R, R_j , ... : rôles atomiques
 - dans les exemples : commence par minuscule (**parentDe**, ...)
 - *pas de rôles complexes*

Syntaxe : concepts complexes

- constructeurs de concepts :

\top	concept universel	“tout”
\perp	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') :

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

où :

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

Syntaxe : exemples

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

Syntaxe : exercices *et solutions*

- étant donné les ensembles

$NomsConcepts = \{Pers, Masculin, Feminine, Riche\}$

$NomsRoles = \{parentDe, soeurDe, frereDe\},$

définir les concepts complexes suivants :

- être une femme $Pers \sqcap Feminine$
- être une mère $Femme \sqcap \exists parentDe. \top$
 - être une mère qui n'a que des garçons $Mere \sqcap \forall parentDe. Masculin$
- être un oncle $Pers \sqcap \exists frereDe. \exists parentDe. \top$
- être un grand-oncle $Pers \sqcap \exists frereDe. \exists parentDe. \exists parentDe. \top$
- être quelqu'un qui a un neveu riche $Pers \sqcap \exists frereDe. \exists parentDe. Riche \sqcup \exists soeurDe. \exists parentDe. Riche$
 - être quelqu'un qui a un oncle riche impossible (pas de converse dans ALC)
- être quelqu'un qui a un seul enfant $Pers \sqcap \exists parentDe. (\forall soeurDe. \perp \sqcap \forall frereDe. \perp)$
 - être quelqu'un qui a deux enfants impossible (pas de restriction de cardinalité dans ALC)

Sémantique

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

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

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

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

Sémantique (suite)

- interprétation de concepts complexes :

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

$$\perp^I = \emptyset$$

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

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

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

$$(\forall R.C)^I = \{a \in \Delta^I : \text{pour tout } b \in \Delta^I, \text{ si } (a, b) \in R^I \text{ alors } b \in C^I\}$$

$$(\exists R.C)^I = \{a \in \Delta^I : \text{il existe } b \in \Delta^I \text{ tel que } (a, b) \in R^I \text{ et } b \in C^I\}$$

Sémantique : quelques équivalences

équivalences sémantiques :

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

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

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

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

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

Proposition (forme normale négative)

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

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

● exemple : $\neg\forall\text{parentDe}.\perp$

(... il manque une équivalence)

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ**
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

Le langage ALCN : ALC + restrictions de cardinalité

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

$$C ::= N_C \mid \top \mid \perp \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \forall N_R.C \mid \exists N_R.C \mid \leq n N_R \mid \geq n N_R$$

où :

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

solution : $\leq n R \sqcap \geq n R$

ALCN : sémantique

- les mêmes interprétations que dans ALC :

- $\mathcal{I} = (\Delta^{\mathcal{I}}, \cdot^{\mathcal{I}})$ telle que

- $\Delta^{\mathcal{I}}$ est un ensemble non-vidé (domaine)
- $(\cdot)^{\mathcal{I}} : \text{NomsConcepts} \rightarrow 2^{\Delta^{\mathcal{I}}}$ (interprétation des concepts)
- $(\cdot)^{\mathcal{I}} : \text{NomsRoles} \rightarrow 2^{\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}}$ (interprétation des rôles)

- interprétation des concepts complexes :

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

$$(\geq n R)^{\mathcal{I}} = \dots$$

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

preuve non triviale (utilise la *bisimulation* entre interprétations)

ALCN : exercices et solutions

- décrire les concepts suivants :

- 1 être une personne avec au moins trois enfants

$$\text{Pers} \sqcap \geq 3 \text{parentDe}$$

- 2 être une personne avec exactement trois enfants

$$\text{Pers} \sqcap \geq 3 \text{parentDe} \sqcap \leq 3 \text{parentDe}$$

- 3 être un cours avec au plus 15 participants dont tous sont des étudiants de master

- concepts primitifs : Cours, EtudiantEnMaster
- rôle primitif : aParticipant

$$\text{Cours} \sqcap \leq 15 \text{aParticipant} \sqcap \forall \text{aParticipant} . \text{EtudiantEnMaster}$$

- 4 être une grand-mère avec une fille qui a deux fils comme seuls enfants

$$\text{Femme} \sqcap \exists \text{parentDe} . (\text{Femme} \sqcap \leq 2 \text{parentDe} \sqcap \forall \text{parentDe} . \text{Masculin})$$

- étendre la définition de la forme normale négative aux restrictions de cardinalité

$$\neg \leq n R = \geq n+1 R$$

- est-ce qu'on peut décrire le concept 'être une mère qui a (au moins) deux fils' ?

il faudrait $\text{Mere} \sqcap \geq 2 \text{parentDe} . \text{Masculin}$, mais ... ALCN ne le permet pas !

ALCQ : restrictions de cardinalité *qualifiées* avec solutions

- grammaire des concepts ('BNF') :

$$C ::= N_C \mid \top \mid \perp \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid \leq n N_R.C \mid \geq n N_R.C$$

où $N_C \in \text{NomsConcepts}$, $N_R \in \text{NomsRoles}$, $n \geq 0$

- $\leq n R.C$ = 'au plus n R-successeurs sont dans C '
- $\geq n R.C$ = 'au moins n R-successeurs sont dans C '
- interprétation :

$$(\leq n R.C)^I = \{a \in \Delta^I : \text{Card}(\{b \in \Delta : (a, b) \in R^I \text{ et } b \in C^I\}) \leq n\}$$

$$(\geq n R.C)^I = \{a \in \Delta^I : \text{Card}(\{b \in \Delta : (a, b) \in R^I \text{ et } b \in C^I\}) \geq n\}$$

- ALCQ est au moins aussi expressif que ALCN :

- $\leq n R \stackrel{\text{def}}{=} \leq n R.\top$
- $\geq n R \stackrel{\text{def}}{=} \geq n R.\top$
- $\exists R.C \stackrel{\text{def}}{=} \geq 1 R.C$
- $\forall R.C \stackrel{\text{def}}{=} \leq 0 R.\neg C$

- ALCQ est **strictement** plus expressif que ALCN
 - il y a des restrictions de cardinalité qualifiées qui ne peuvent pas être exprimées dans ALCN

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox**
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

ABox = *assertion box*

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

Différence avec les bases de données

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

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox**
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

TBox générale

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

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

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

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

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

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

- exercices :

- 1 exprimer par un axiome que le 2nd argument de la relation R a la propriété C ('restriction du codomaine')
 - en FOL: $\forall x \forall y (R(x, y) \rightarrow C(y))$ en ALC: $\top \sqsubseteq \forall R.C$
- 2 exprimer par un axiome que le 1er argument de la relation R a la propriété C ('restriction du domaine')
 - en FOL : $\forall x \forall y (R(x, y) \rightarrow C(x))$ en ALC: $\exists R.\top \sqsubseteq C$

TBox acyclique

- les axiomes d'inclusion peuvent être éliminés :
 - remplacer $C \sqsubseteq D$ par $C \equiv D \sqcap A$, où A est nouveau
 - équivalent à la TBox d'origine \mathcal{T} p.r. au langage de \mathcal{T}

$$\mathcal{T} \cup \{C \sqsubseteq D\} \models C \text{ ssi } \mathcal{T} \cup \{C \equiv D \sqcap A\} \models C \text{ si } A \text{ pas dans } C$$

- $C \equiv D$ est une **définition** ssi C est atomique
 - $\text{etreOncle} \equiv \text{Masculin} \sqcap \exists \text{frereDe} . \exists \text{parentDe} . \top$

\Rightarrow problème : $A \equiv A \sqcup C$ 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 : $A \equiv C, A \equiv C' \in \mathcal{T}$ et $C \neq C'$
 - 3 pas de cycles
 - interdit : $A_1 \equiv \dots A_2 \dots, A_2 \equiv \dots A_3 \dots, \dots, A_n \equiv \dots A_1 \dots$
 - si $A \equiv C \in \mathcal{T}$ alors A est un concept non-primitif

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

\Rightarrow intéressant pour le calcul

Exemple d'une KB généalogique

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

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

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement**
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

Tâches de raisonnement sur les concepts

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

- exemples :
 - $\text{Mere} \sqcap \forall \text{parentDe}.\perp$ est satisfaisable
 - $\forall \text{parentDe}.C \sqcap \exists \text{parentDe}.\neg C$ est insatisfaisable
 - $\forall \text{parentDe}.C \sqcap \forall \text{parentDe}.\neg C$ 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 p.r.à une TBox

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

- exemple :
 - $\text{Mere} \sqcap \forall \text{parentDe} . \perp$ 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)

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

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

Comment se débarrasser 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 :**

$$\mathcal{T}_{gen} = \left\{ \begin{array}{lll} \text{Femme} & \equiv & \text{Pers} \sqcap \text{Feminine}, \\ \text{Homme} & \equiv & \text{Pers} \sqcap \text{Masculin}, \\ \text{Mere} & \equiv & \text{Femme} \sqcap \exists \text{parentDe.Pers}, \\ \text{Pere} & \equiv & \text{Homme} \sqcap \exists \text{parentDe.Pers}, \\ \text{Parent} & \equiv & \text{Mere} \sqcup \text{Pere}, \\ \text{MereSansFille} & \equiv & \text{Mere} \sqcap \forall \text{parentDe.}\neg \text{Femme} \end{array} \right\}$$

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

Comment se débarrasser d'une TBox acyclique (suite),

avec solution

Proposition

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

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

$$A_1 \equiv \exists R_1.A_2 \sqcap \exists R_2.A_2$$

$$A_2 \equiv \exists R_1.A_3 \sqcap \exists R_2.A_3$$

$$A_3 \equiv \exists R_1.A_4 \sqcap \exists R_2.A_4$$

Tâches de raisonnement p.r.à une ABox

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

- satisfaisabilité p.r.à une TBox :
 - \mathcal{A} satisfaisable p.r.à \mathcal{T} ssi il existe \mathcal{I} tel que $\mathcal{I} \models \mathcal{T}$ et $\mathcal{I} \models \mathcal{A}$
 - 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
 - C est satisfaisable ssi $\{C(a)\}$ est satisfaisable, pour un individu a quelconque
- inférence de propriétés :
 - $\mathcal{T} \cup \mathcal{A} \models C(a)$ ssi pour tout \mathcal{I} , si $\mathcal{I} \models \mathcal{T}$ et $\mathcal{I} \models \mathcal{A}$ alors $\mathcal{I} \models C(a)$
 - exemple :
 - $\mathcal{T}_{gen} \cup \{\text{GrandMere}(\text{Alice})\} \models \text{Mere}(\text{Alice})$
 - $\mathcal{T} \cup \mathcal{A} \models C(a)$ ssi $\mathcal{A} \cup \{\neg C(a)\}$ insatisfaisable p.r.à \mathcal{T}
- requête :
 - trouver tous les a tel que $\mathcal{T} \cup \mathcal{A} \models C(a)$
- réalisation :
 - calculer les noms de concept les plus spécifiques de la TBox pour un individu donné (cf. la classification)
 - Bob est instance de Pers, Masculin, Pere

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre**
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

Rappel : la logique du premier ordre FOL avec solutions

- langage :
 - variables objet x, y, \dots
 - prédicats : $P(t_1, \dots, t_n)$
 - 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 $\mathcal{I} = (\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$:
 - 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 :
 - $(\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}}) \models_{FOL} P(t_1, \dots, t_n)$ ssi $((t_1)^{\mathcal{I}}, \dots, (t_n)^{\mathcal{I}}) \in (P)^{\mathcal{I}}$
 - $(\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}}) \models_{FOL} \forall x \varphi$ ssi $(\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}_x}) \models_{FOL} \varphi$ pour toute variante \mathcal{I}_x de \mathcal{I} en x
 - ...
- formule φ est *valide* ssi $(\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}}) \models_{FOL} \varphi$, pour tout $(\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$
- formule φ est *satisfaisable* ssi ... il existe $(\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$ t.q.

Traduction de ALC en FOL avec solution

concept $C \mapsto$ formule $\Phi(C, x)$ de FOL

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

$$\Phi(A, x) = A(x)$$

$$\Phi(\top, x) = \top$$

$$\Phi(\perp, x) = \perp$$

$$\Phi(\neg C, x) = \neg \Phi(C, x)$$

$$\Phi(C \sqcap D, x) = \Phi(C, x) \wedge \Phi(D, x)$$

$$\Phi(C \sqcup D, x) = \Phi(C, x) \vee \Phi(D, x)$$

$$\Phi(\forall R.C, x) = \forall y (R(x, y) \rightarrow \Phi(C, y))$$

$$\Phi(\exists R.C, x) = \exists y (R(x, y) \wedge \Phi(C, y))$$

Traduction de ALC en FOL (suite)

Proposition

Pour tout concept C et pour toute interprétation $(\Delta^I, (.)^I)$:

$$C^I = \{a \in \Delta^I : \mathcal{I} \models_{FOL} \Phi(C, a)\}$$

démonstration par induction sur la forme de C ; on montre :

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

- ① si C est un concept atomique A alors :

$$\begin{aligned} a \in A^I & \text{ ssi } \mathcal{I} \models_{FOL} A(a) \\ & \text{ ssi } \mathcal{I} \models_{FOL} \Phi(A, a) \end{aligned}$$

- ② si C est de la forme \top alors ...

- ③ si C est de la forme \perp alors ...

- ④ si C est de la forme $\neg D$ alors :

$$\begin{aligned} a \in (\neg D)^I & \text{ ssi } a \notin D^I \\ & \text{ ssi } \mathcal{I} \not\models_{FOL} \Phi(D, a) \quad (\text{par H.I.}) \\ & \text{ ssi } \mathcal{I} \models_{FOL} \Phi(\neg D, a) \end{aligned}$$

- ⑤ si ...

Traduction de ALC en FOL : exercices

$$\Phi(A, x) = A(x)$$

$$\Phi(\top, x) = \top$$

$$\Phi(\perp, x) = \perp$$

$$\Phi(\neg C, x) = \neg\Phi(C, x)$$

$$\Phi(C \sqcap D, x) = \Phi(C, x) \wedge \Phi(D, x)$$

$$\Phi(C \sqcup D, x) = \Phi(C, x) \vee \Phi(D, x)$$

$$\Phi(\forall R.C, x) = \forall y (R(x, y) \rightarrow \Phi(C, y))$$

$$\Phi(\exists R.C, x) = \exists y (R(x, y) \wedge \Phi(C, y))$$

- traduire en FOL :

- $\text{Pers} \sqcap \exists \text{parentDe} . \exists \text{parentDe} . \text{Masculin}$

- $(\forall \text{parentDe} . \text{Masculin}) \sqcap (\forall \text{parentDe} . \neg \text{Masculin})$

- montrer que la traduction est satisfaisable : donner une interprétation de FOL $\mathcal{I} = (\Delta^{\mathcal{I}}, (\cdot)^{\mathcal{I}})$ où elle est vraie

Traduction de ALCN en FOL : exercices

- compléter la traduction de ALCN en FOL

$$\Phi(\geq 2 R, x) = \dots$$

$$\Phi(\geq n R, x) = \dots$$

$$\Phi(\leq n R, x) = \dots$$

- traduire en FOL

- $\text{Pers} \sqcap \leq 2 \text{parentDe}$

Traduction de ALCN en FOL : solution

$$\Phi(\leq 2 R, x) = \forall y_1 \forall y_2 \forall y_3 \left((R(x, y_1) \wedge R(x, y_2) \wedge R(x, y_3)) \rightarrow \right. \\ \left. (y_1 = y_2 \vee y_1 = y_3 \vee y_2 = y_3) \right)$$

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

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

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales**
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement

Logiques modales : introduction

- même sémantique que les DL : interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, (.)^{\mathcal{I}})$ avec

- $\Delta^{\mathcal{I}}$ un ensemble non-vide
- $(.)^{\mathcal{I}} : \text{NomsConcepts} \rightarrow 2^{\Delta^{\mathcal{I}}}$
- $(.)^{\mathcal{I}} : \text{NomsRoles} \rightarrow 2^{\Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}}$

- mais reformulée : *modèle de Kripke* $M = (W, \mathcal{R}, V)$, avec :

- W ensemble quelconque non-vide
 - $w \in W$ = "monde possible" = description complète du monde
- $\mathcal{R} : \text{Indices} \rightarrow 2^{W \times W}$, pour $\text{Indices} = \{i, j, \dots\}$
 - \mathcal{R}_i = "relation d'accessibilité"
- $V : \text{Atomes} \rightarrow 2^W$, pour $\text{Atomes} = \{p, q, \dots\}$
 - $V(p)$ = "valuation de la variable propositionnelle p "

- intuition diffère :

DL		ML	
$\Delta^{\mathcal{I}}$	objets du monde	W	mondes possibles
$a \in \mathbf{A}^{\mathcal{I}}$	"a a la propriété A"	$w \in V(p)$	"p est vrai à w"

- applications diffèrent :

- Atomes = faits du monde ('il pleut', ...)
- Indices = agents/actions/systèmes normatifs/...

Logiques modales : applications

- but : ajouter à (la syntaxe de) la logique propositionnelle des *modalités*
 - ‘modalité’ : modifie le sens d’une proposition
 - beaucoup de modalités existent :
 - “il est nécessaire que φ ” / “il est possible que φ ” (~1912)
 - “ φ sera toujours vrai” / “ φ sera vrai parfois” (~1960)
 - “ φ sera vrai après (tout) exécution de l’action α ” (~1980)
 - “il est obligatoire que φ ” / “il est permis que φ ” (~1970)
 - “l’agent i croit que φ ” (~1960)
 - “l’agent i sait que φ ” (~1960)
- pertinentes pour :
 - 1 vérification des programmes : logique temporelle
 - 2 systèmes distribués : logique temporelle, épistémique
 - 3 systèmes multi-agents : logique épistémique, doxastique, temporelle, dynamique, déontique
 - 4 sémantique du langage naturel

Logiques modales : points communs

- opérateurs *non vérifonctionnels*
 - la valeur de vérité de “il est nécessaire que φ ” n’est pas fonction de la valeur de vérité de φ
- dualité des opérateurs : comme \forall et \exists
 - φ est nécessaire ssi $\neg\varphi$ est impossible
 - φ sera toujours vrai ssi il n’est pas le cas que φ est vrai parfois
 - φ sera vrai après toute exécution de α ssi il n’y a pas d’exécution possible de α après laquelle φ est faux
 - φ est obligatoire ssi $\neg\varphi$ n’est pas permis
- notation générique, pour $i \in Indices$:
 - $\Box_i\varphi$ = “ φ est nécessaire p.r.à i ”
 - $\Diamond_i\varphi$ = “ φ est possible p.r.à i ”

Logiques modales : syntaxe

- grammaire du langage de la logique modale \mathcal{L}_{ML} :

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid (\varphi \rightarrow \varphi) \mid \Box_i\varphi \mid \Diamond_i\varphi$$

pour $p \in \text{Atomes}$ et $i \in \text{Indices}$

- $\Box_i\varphi$ = “ φ est nécessaire p.r.à i ”
- $\Diamond_i\varphi$ = “ φ est possible p.r.à i ”

Logiques modales : deux caractérisations

- logique = langage \mathcal{L}_{ML} plus sous-ensemble de \mathcal{L}_{ML} : les ‘théorèmes’/‘validités’
 - alternativement : logique = langage \mathcal{L}_{ML} plus
 - le sous-ensemble de \mathcal{L}_{ML} des formules consistantes
 - l’ensemble des couples (φ, ψ) tel que ψ est conséquence logique de φ
- le sous-ensemble peut être caractérisé de deux manières :
 - sémantiquement = par des interprétations (en ML : ‘modèles’)
 - 1 φ est valide = φ est vrai dans chaque modèle
 - 2 φ est satisfaisable = φ est vrai dans au moins un modèle
 - axiomatiquement = par des axiomes et des règles d’inférence
 - 1 φ est un théorème = φ est démontrable à partir des axiomes via les règles d’inférence
 - 2 φ est consistant = on ne peut pas démontrer $\neg\varphi$
- la communauté DL considère uniquement des caractérisations sémantiques

Logiques modales : sémantique

- pour un modèle $M = (W, \mathcal{R}, V)$ donné :
chaque (M, w) avec $w \in W$ est un *modèle pointé*
- vérité d'une formule φ de \mathcal{L}_{ML} dans un point w du modèle M :

$M, w \Vdash p$ si $w \in V(p)$, pour p atomique

$M, w \Vdash \neg\varphi$ si $M, w \not\Vdash \varphi$

$M, w \Vdash \varphi \wedge \psi$ si $M, w \Vdash \varphi$ et $M, w \Vdash \psi$

...

$M, w \Vdash \Box_i \varphi$ si pour tout v tel que $(w, v) \in \mathcal{R}(i)$, $M, v \Vdash \varphi$

$M, w \Vdash \Diamond_i \varphi$ si il existe un v et $(w, v) \in \mathcal{R}(i)$ tel que $M, v \Vdash \varphi$

Logiques modales : caractérisations sémantiques

- **vérité** d'une formule dans M ('vérité globale') :

$$M \Vdash \varphi \text{ ssi } M, w \Vdash \varphi \text{ pour tout } w \in W$$

- **validité** d'une formule **dans une classe de modèles** C :

$$\Vdash_C \varphi \text{ ssi pour tout modèle } M \in C, M \Vdash \varphi$$

- **conséquence logique** dans une classe de modèles C :

$$\varphi \Vdash_C \psi \text{ ssi pour tout modèle } M \in C, \text{ si } M \Vdash \varphi \text{ alors } M \Vdash \psi$$

- donc φ est valide dans C ssi $\top \Vdash_C \varphi$
- exemple de conséquence logique : $\varphi \Vdash_C \Box_i \varphi$
 - attention : $\not\vdash_C \varphi \rightarrow \Box_i \varphi$
 - pas de théorème de déduction en ML !
 - à moins que C est tel que...
- cas particulier : la classe de *tous* les modèles de Kripke \mathcal{K}
 - "logique (multi)modale \mathcal{K} "

Traduction d'un concept ALC en logique modale \mathcal{K}

- fonction de traduction :

$$\mu(A) = A$$

si A est un concept atomique

$$\mu(\top) = \top$$

$$\mu(\perp) = \perp$$

$$\mu(\neg C) = \neg\mu(C)$$

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

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

$$\mu(\forall R.C) = \Box_R \mu(C)$$

$$\mu(\exists R.C) = \Diamond_R \mu(C)$$

- un concept C est satisfaisable ssi $\not\models_{\mathcal{K}} \neg\mu(C)$
- autres tâches de raisonnement
 - le concept C subsume D
 - ssi $\models_{\mathcal{K}} \mu(\neg(D \sqcap \neg C))$
 - ssi $\models_{\mathcal{K}} \mu(D) \rightarrow \mu(C)$
 - ...

Traduction d'une TBox ALC en logique modale \mathcal{K}

- traduction d'une TBox :

$$\mu(\mathcal{T}) = \bigwedge_{C \sqsubseteq D \in \mathcal{T}} (\mu(C) \rightarrow \mu(D))$$

- un concept C est satisfaisable p.r.à \mathcal{T} ssi $\mu(\mathcal{T}) \not\models_{\mathcal{K}} \neg\mu(C)$
- autres tâches de raisonnement
 - le concept C subsume D p.r.à \mathcal{T} ssi ...
 - ...

Traduction d'une ABox ALC en logique modale \mathcal{K}

- requiert une extension non-standard –mais ‘gratuite’ en termes de complexité– de la logique multimodale \mathcal{K}
 - nominaux = noms (‘indices’) de mondes
 - $M, w \Vdash a$ si a est le nom de w
 - $M, w \Vdash @_a \varphi$ si a est le nom de $v \in W$ et $M, v \Vdash \varphi$
- traduction d'une ABox :

$$\mu(\mathcal{A}) = \left(\bigwedge_{C(a) \in \mathcal{A}} @_a \mu(C) \right) \wedge \left(\bigwedge_{R(a,b) \in \mathcal{A}} @_a \diamond Rb \right)$$

- une ABox \mathcal{A} est satisfaisable ssi $\mu(\mathcal{A}) \not\models_{\mathcal{K}} \perp$
- une ABox \mathcal{A} est satisfaisable p.r.à une TBox \mathcal{T} ssi $\mu(\mathcal{A}) \wedge \mu(\mathcal{T}) \not\models_{\mathcal{K}} \perp$
- autres tâches de raisonnement
 - inférence de propriétés : $\mathcal{T} \cup \mathcal{A} \models C(a)$
ssi $\mu(\mathcal{T}) \wedge \mu(\mathcal{A}) \models C(a)$
 - ...

Logiques modales au-delà de \mathcal{K}

- selon l'application, les opérateurs doivent avoir des propriétés différentes
 - “ i sait que φ ” implique que φ est vrai
 - “ i croit que φ ” n'implique pas que φ est vrai
- c'est pris en compte sémantiquement en restreignant la classe de modèles C qu'on considère :
 - en logique de la connaissance $\Box_i\varphi \rightarrow \varphi$ doit être valide dans chaque modèle
 - tandis que la logique de la croyance doit avoir des modèles pointés où $\Box_i\varphi \rightarrow \varphi$ est faux
- c'est pris en compte axiomatiquement en rajoutant des axiomes

Logique dynamique (logique de l'action)

- lecture:
 - $\Box_a \varphi$ = “ φ est vrai après toute exécution de l'action a ”
 - $\Diamond_a \varphi$ = “il existe une exécution de l'action a telle que φ est vrai après”
- logique \mathcal{K}
 - une action peut être non-déterministe
- cas particulier: chaque action est déterministe
 - modèles où chaque relation $\mathcal{R}(a)$ est une fonction
 - $\Diamond_a \varphi \rightarrow \Box_a \varphi$ devient valide (\Rightarrow axiome)
- les actions peuvent être complexes :
 - $a; b$ = exécution séquentielle de a et ensuite b
 - $a \cup b$ = choix non-déterministe entre l'exécution de a et l'exécution de b
 - $\varphi?$ = test si la formule φ est vraie
 - ...

Logique temporelle

- lecture:
 - $\Box\varphi$ = “ φ est vrai à tout instant successeur”
 - $\Diamond\varphi$ = “il y a un instant successeur où φ est vrai”

(un seul indice \Rightarrow omis)
- doit être valide :
 - $\Diamond\top$

\Rightarrow axiome (hypothèse : le temps ne s'arrête jamais)
- modèles : la relation d'accessibilité est *sérielle*
 - chaque monde possible a au moins un successeur
 - par contre le futur peut être non-déterministe
 - axiome pour le futur déterministe : $\Diamond\varphi \rightarrow \Box\varphi$
- d'autres opérateurs :
 - $\Box^{-1}\varphi$ = “ φ est vrai à tout instant prédecesseur”
 $\Diamond^{-1}\varphi = \dots$
 - doivent être valides : $\varphi \rightarrow \Box\Diamond^{-1}\varphi, \varphi \rightarrow \Box^{-1}\Diamond\varphi$
 - $\Box^+\varphi$ = “ φ est vrai toujours dans le futur”
 $\Diamond^+\varphi = \dots$
 - doivent être valides : $\Box^+\varphi \rightarrow \Box\varphi, \Diamond^+\Diamond^+\varphi \rightarrow \Diamond^+\varphi, \dots$
 - $\mathcal{R}(\Box^+)$ fermeture transitive de $\mathcal{R}(\Box)$

Logique aléthique

- aléthique = nécessaire
- lecture:
 - $\Box\varphi$ = “il est nécessaire que φ ”
 - $\Diamond\varphi$ = “il est possible que φ ”

(un seul indice \Rightarrow omis)

- doit être valide :
 - $\Box\varphi \rightarrow \varphi$
 - $\Box\varphi \rightarrow \Box\Box\varphi$
 - $\varphi \rightarrow \Box\Diamond\varphi$

\Rightarrow axiomes

- modèles : la relation d'accessibilité \mathcal{R} est une relation d'équivalence
 - réflexive, transitive et symétrique

Logique déontique

- déontique = relatif aux obligations
- lecture:
 - $\Box\varphi$ = “il est obligatoire que φ ”
 - $\Diamond\varphi$ = “il est permis que φ ”(un seul indice \Rightarrow omis)
- doit être valide :
 - $\neg(\Box\varphi \wedge \Box\neg\varphi)$ \Rightarrow axiome
- modèles : la relation d'accessibilité \mathcal{R} est sérielle

Logique doxastique

- doxa = croyance
- lecture:
 - $\Box_i \varphi$ = "l'agent i croit que φ "
 - $\Diamond_i \varphi$ = "il es compatible avec les croyances de i que φ "
- doit être valide :
 - $\neg(\Box_i \varphi \wedge \Box_i \neg \varphi)$
 - $\Box_i \varphi \rightarrow \Box_i \Box_i \varphi$ ('introspection positive')
 - $\Diamond_i \varphi \rightarrow \Box_i \Diamond_i \varphi$ ('introspection négative')

\Rightarrow axiomes (N.B. : il s'agit d'une idéalisation)
- modèles : les relations d'accessibilité $\mathcal{R}(i)$ sont *sérielles*, *transitives* et *euclidéennes*
 - *sérialité* : pour tout $w \in W$ il existe $v \in W$ tel que $(w, v) \in \mathcal{R}(i)$
 - *euclidéanité* : si $(w, v) \in \mathcal{R}(i)$ et $(w, u) \in \mathcal{R}(i)$ alors $(v, u) \in \mathcal{R}(i)$

Logiques épistémique

- épistémé = connaissance
- lecture:
 - $\Box_i \varphi$ = "l'agent i sait que φ "
 - $\Diamond_i \varphi$ = "il es compatible avec les connaissances de i que φ "
- doit être valide :
 - $\Box_i \varphi \rightarrow \varphi$
 - $\Box_i \varphi \rightarrow \Box_i \Box_i \varphi$ ('introspection positive')
 - $\Diamond_i \varphi \rightarrow \Box_i \Diamond_i \varphi$ ('introspection négative')
- ⇒ axiomes
- modèles : la relation d'accessibilité \mathcal{R} est une relation d'équivalence
 - reflexive, transitive et euclidéenne
 - même chose que : réflexive, transitive et symétrique (v.s.)

Combinaisons de logiques modales

- épistémique + doxastique
 - “ i croit que j sait si φ est vrai”
 - doivent être valides :
 - savoir implique croire
 - croire implique croire savoir (?)
 - ...
 - est-ce qu'on peut définir la connaissance à partir de la croyance ?
 - problème non résolu depuis 2500 ans
 - Aristote : i sait que $\varphi = i$ croit que φ , φ est vrai, et i a une justification pour φ
 - paradoxe de Gettier [1963]
- épistémique + dynamique
 - “ i sait que φ est faux maintenant, mais sera vrai après l'exécution de α ”
- épistémique + temporel
- épistémique + déontique
 - “il est permis que i sait que φ ”
 - “ i ne sait pas que φ est interdit”
- ...

Traduction de la logique modale \mathcal{K} en ALC

- traduction d'une formule modale φ en un concept :

$$\kappa(p) = A_p \quad \text{si } p \text{ est un atome}$$

$$\kappa(\neg\varphi) = \neg\kappa(\varphi)$$

$$\kappa(\varphi \wedge \psi) = \kappa(\varphi) \sqcap \kappa(\psi)$$

$$\kappa(\varphi \vee \psi) = \kappa(\varphi) \sqcup \kappa(\psi)$$

$$\kappa(\varphi \rightarrow \psi) = \neg\kappa(\varphi) \sqcup \kappa(\psi)$$

$$\kappa(\Box_i\varphi) = \forall R_i.\kappa(\varphi)$$

$$\kappa(\Diamond_i\varphi) = \exists R_i.\kappa(\varphi)$$

- $\models_{\mathcal{K}} \varphi$ (φ est valide dans la classe de tous les modèles de Kripke \mathcal{K})
ssi le concept $\neg\kappa(\varphi)$ est insatisfaisable en ALC
- $\varphi \models_{\mathcal{K}} \psi$ (φ a comme conséquence logique ψ dans \mathcal{K})
ssi $\neg\kappa(\psi)$ est in insatisfaisable en ALC p.r.à la TBox
 $\{\top \sqsubseteq \kappa(\varphi)\}$
- traduction d'autres logiques modales dans des DL : utilise des inclusions de rôles pour tenir compte des propriétés de la relation d'accessibilité
 - $R \circ R \sqsubseteq R$ pour la transitivité, etc.

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ**
- 9 Mécanismes de raisonnement

Le langage SROIQ

- essentiellement OWL DL

- ALCQ plus :

- rôle universel U
- hiérarchies de rôle

$\text{frereDe} \circ \text{parentDe} \sqsubseteq \text{oncleDe}$

- nominaux = concepts particuliers désignant un seul individu

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

$\{\text{Alice}\} \approx \{\text{Anne}\}$

... donc pas de UNA !

$\{\text{Alice}\} \not\approx \{\text{Anne}\}$

équivalent : $\{\text{Alice}\} \sqcap \{\text{Anne}\} \sqsubseteq \perp$

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

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

$\text{Mere} \sqsubseteq \{\text{Alice}\} \sqcup \{\text{Anne}\}$... ne peut pas être exprimé en ALC !

- rôles inverses

parentDe^-

- *Self*

$\neg \exists \text{parentDe. Self}$

Le langage SROIQ

- grammaire :

rôles : $R ::= U \mid N_R \mid N_R^-$

concepts : $C ::= N_C \mid \{N_I\} \mid \top \mid \perp \mid \neg C \mid C \sqcap C \mid C \sqcup C \mid$
 $\forall R.C \mid \exists R.C \mid \leq n R.C \mid \geq n R.C \mid \exists R.Self$

assertions : $\alpha ::= C(N_I) \mid R(N_I, N_I) \mid N_I \approx N_I \mid N_I \neq N_I$

axiomes : $\tau ::= C \sqsubseteq C \mid C \equiv C \mid$

$R \sqsubseteq R \mid R \equiv R \mid R \circ R \sqsubseteq R \mid Disjoint(R, R)$

- en plus :

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

Outline

- 1 Le langage ALC
- 2 Les langages ALCN et ALCQ
- 3 Structure et propriétés des ABox
- 4 Structure et propriétés des TBox
- 5 Tâches de raisonnement
- 6 Relation avec la logique du premier ordre
- 7 Les logiques modales
- 8 Le langage SROIQ
- 9 Mécanismes de raisonnement**

Décidabilité et complexité

- le problème de satisfaisabilité d'un concept est décidable :
 - PSPACE complet si la TBox est acyclique
 - EXPTIME complet en général
- ⇒ suite : procédure de décision via tableaux pour la satisfaisabilité d'une ABox
- ⇒ 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 $S \cup \{\mathcal{A}\}$
 - $\mathcal{A}[X] = “X \text{ est un sous-ensemble de } \mathcal{A}”$
 - format des règles :

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

- si $X \subseteq \mathcal{A}$ alors ajouter Y_i à \mathcal{A}
- applicable si $Y_1 \not\subseteq \mathcal{A}, \dots$ et $Y_n \not\subseteq \mathcal{A}$

Règles de tableau pour ALC

- règle pour \sqcap :

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

- règle pour \sqcup :

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

- règle pour \exists :

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

b = nouveau individu

- règle pour \forall :

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

Tableaux : contradictions manifeste

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

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

Sinon \mathcal{A} est *ouvert*.

- exemple :

$$\mathcal{A} = \{ \text{Pers} \sqcap (\exists \text{parentDe.Pers}) \sqcap \forall \text{parentDe}.\perp(a) \}$$

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

Raisonnement : exercice complet (1)

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

(tâche = inférence de propriétés)

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

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

- ② éliminer \sqsubseteq :

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

\Rightarrow TBox acyclique !

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

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

- ④ mise en forme normale négative :

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

- ⑤ construction d'un tableau ...

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)

⇒ contradiction manifeste

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

⇒ 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 (2)

$\{((\exists R.B_1) \sqcap (\exists R.B_2) \sqcap \forall R.(\neg B_1 \sqcup \neg B_2))(a)\}$ 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. $\mathcal{I} \models \mathcal{A}_i$.
- 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
- rôle reflexif : on ajoute une règle supplémentaire pour \forall

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

- rôle transitif : ...
- restrictions de cardinalité qualifiées
 - indécidable si combiné avec intersection de rôles !
 - indécidable si combiné avec rôles transitifs !
- constructeur 'un-parmi a_1, \dots, a_n ' : $\{a_1, \dots, a_n\}$
 - $(\{a_1, \dots, a_n\})^I = \{a_1^I, \dots, a_n^I\}$
 - exemple : Pers \equiv {Alice, Bob, Charles}
- domaines concrètes
 - nombres entiers : Adulte \equiv Pers \sqcap \exists ageDe. \geq 18
 - ...

Conclusion sur la partie DL et ML du cours

- idée commune : représenter + raisonner
 - classes de complexité : P, NP, PSPACE, EXPTIME
- DL : choix de la logique dépend de l'expressivité souhaitée
 - quelles restrictions du langage pour quel domaine ?
 - recherches actuelles :
 - langages d'interrogation (*query languages*) à complexité basse
 ⇒ restrictions drastiques du langage : "DL-lite"
 - *révision* d'une TBox ; *mise à jour* d'une ABox
 - réparation d'une ABox devenue insatisfaisable p.r.à une TBox
 - alignement de deux TBox : <http://ontologymatching.org/>
 (IRIT : Cassia Trojahn)
- ML : choix de la logique dépend du concept modélisé
 - connaissance : $\models \Box_i \varphi \rightarrow \varphi$; croyance : $\not\models \Box_i \varphi \rightarrow \varphi$
 - recherches actuelles :
 - réduction de la complexité par des restrictions du langage (cf. DL) (IRIT : Andreas Herzig)
 - logiques épistémico-dynamiques (IRIT : Hans van Ditmarsch)
 - logiques doxastiques cognitivement adéquates : croyances explicites vs. croyances implicites (IRIT : Emiliano Lorini)