UPMC
ANNÉE 2017-2018

LRC - Annexe n°7

M1 ANDROIDE
M1 DAC

Logiques de description

Ce document récapitule la syntaxe de $\mathcal{FL}^-$, $\mathcal{ALC}$ et des principales extensions de cette dernière. Pour ceux approfondir le sujet, il est possible de se référer à `http://dl.kr.org/` qui centralise un certain nombre de ressources sur le sujet, dont en particulier `http://www.cs.man.ac.uk/~ezolin/dl/` qui donne les différentes extensions de $\mathcal{ALC}$ et les résultats de complexité qui s'y rapportent.

1 $\mathcal{FL}^-$

1.1 Alphabet

- Concepts atomiques : A, B, C, ...
- Rôles atomiques : R, S, U, V, ...
- Symboles : $\{\sqcap, \exists, \forall, .\}$
- Instances de concepts : a, b, ...

1.2 Base de connaissances

- TBox - Axiomes terminologiques :
  Définitions : $C \equiv D$
  Subsomptions : $C \sqsubseteq D$
- ABox - Assertions :
  Assertions de concepts : $a : C$
  Assertions de rôles : $\langle a, b\rangle : R$

1.3 Grammaire

concept : := ⟨concept atomique⟩
| ⟨concept⟩ $\sqcap$ ⟨concept⟩
| $\exists$ ⟨rôle atomique⟩
| $\forall$ ⟨rôle atomique⟩.⟨concept⟩
| $\forall$ ⟨rôle atomique⟩.⟨instance⟩

1.4 Sémantique de $\exists$ et $\forall$

Etant donné une interprétation $\mathcal{I} = (\Delta^\mathcal{I}, .^\mathcal{I})$, on a :

- $(\exists R)^\mathcal{I} = \{x \in \Delta^\mathcal{I} | \exists y, (x, y) \in R^\mathcal{I}\}$
  Exemple - avoir au moins un enfant : $\exists$`a_enfant`

- $(\forall R.C)^\mathcal{I} = \{x \in \Delta^\mathcal{I} | \forall y, (x, y) \in R^\mathcal{I} \rightarrow y \in C^\mathcal{I}\}$
  Exemple - avoir uniquement des enfants humains (mais ne pas nécessairement en avoir) : $\forall$`a_enfant.Humain`

2 $\mathcal{ALC}$

2.1 Alphabet

On ajoute les symboles $\sqcup, \neg, \top$ et $\bot$

- Concepts atomiques : A, B, C, ...
- Rôles atomiques : R, S, U, V, ...
- Symboles : $\{\sqcup, \sqcap, \exists, \forall, \neg, \top, \bot, .\}$
- Instances de concepts : a, b, ...

2.2 Grammaire

$$
\begin{array}{ll}
\text{concept} ::= & \langle \text{concept atomique} \rangle \\
& \mid \top \\
& \mid \bot \\
& \mid \neg\langle \text{concept} \rangle \\
& \mid \langle \text{concept} \rangle \sqcap \langle \text{concept} \rangle \\
& \mid \langle \text{concept} \rangle \sqcup \langle \text{concept} \rangle \\
& \mid \exists\ \langle \text{rôle} \rangle.\langle \text{concept} \rangle \\
& \mid \forall\ \langle \text{rôle} \rangle.\langle \text{concept} \rangle
\end{array}
$$

2.3 Sémantique de $\exists$ et $\forall$

Etant donné une interprétation $\mathcal{I} = (\Delta^{\mathcal{I}}, .^{\mathcal{I}})$, on a :

- $(\exists R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists y, (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\}$
  Exemple 1 - avoir au moins un enfant qui est humain : $\exists$`a_enfant.Humain`
  Exemple 2 - ne pas avoir d'enfants qui sont humains (on peut toutefois en avoir qui ne soient pas humains) : $\neg\exists$`a_enfant.Humain`

- $(\forall R.C)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \forall y, (x,y) \in R^{\mathcal{I}} \rightarrow y \in C^{\mathcal{I}}\}$, comme pour $\mathcal{FL}^-$
  Exemple 1 - avoir uniquement des enfants humains (mais ne pas nécessairement en avoir) : $\forall$`a_enfant.Humain`
  Exemple 2 - ne avoir pas uniquement des enfants humains (ie avoir au moins un enfant qui ne soit pas humain) : $\neg\forall$`a_enfant.Humain`

3 Quelques extensions de $\mathcal{ALC}$

3.1 Rôles inverses $\mathcal{I}$

Grammaire : Si R est un rôle, R^{-1} est un rôle.
Sémantique : $(R^{-1})^{\mathcal{I}} = \{(x,y) \in \Delta^{\mathcal{I}} \times \Delta^{\mathcal{I}}, (y,x) \in R^{\mathcal{I}}\}$
Exemple - avoir uniquement des parents humains (mais ne pas nécessairement en avoir) : $\forall$`a_enfant`$^{-1}$`.Humain`

3.2 Restrictions de cardinalité $\mathcal{N}$

Grammaire : Si R est un rôle, $\geq n\ R$ et $\leq n\ R$ sont des concepts.
Note : $\geq n\ R$, resp. $\leq n\ R$, est la notation de `http://www.cs.man.ac.uk/~ezolin/dl/`. On peut aussi le noter $\exists^{\geq n}$, resp. $\exists^{\geq n}R$, comme dans le cours.
Sémantique :
— $(\geq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | Card(\{y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}}\} \geq n\}$
— $(\leq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | Card(\{y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}}\} \leq n\}$
Exemple 1 - avoir moins de 3 enfants (ie 3 ou -) : ≤ 3 `a_enfant`
Exemple 2 - avoir plus de 2 enfants (ie 2 ou +) : ≥ 2 `a_enfant`

3.3 Restrictions de cardinalité qualifiées $\mathcal{Q}$

Grammaire : Si C est un concept et R est un rôle, $\geq n\ R.C$ et $\leq n\ R.C$ sont des concepts.
Sémantique :
— $(\geq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | Card(\{y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \geq n\}$
— $(\leq n\ R)^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | Card(\{y \in \Delta^{\mathcal{I}}, (x,y) \in R^{\mathcal{I}} \wedge y \in C^{\mathcal{I}}\} \leq n\}$
Exemple 1 - avoir plus de 2 filles (enfants qui sont des femmes) : ≥ 2 `a_enfant.Femme`
Exemple 2 - avoir moins de 3 filles (mais on peut avoir aucun fils) : ≤ 3 `a_enfant.Femme`

3.4 Concepts nominaux $\mathcal{O}$

Grammaire : Si $a_1, \ldots, a_n$ sont des instances de concept (ou constantes), $\{a_1, \ldots, a_n\}$ est un concept.
Sémantique :
— $\{a_1, \ldots, a_n\}^{\mathcal{I}} = \{x \in \Delta^{\mathcal{I}} | \exists i \in \{1, \ldots, n\}, x = a_i^{\mathcal{I}}\}$
Exemple 1 - les parents de Robert (avoir Robert comme enfant) : `a_enfant.{Robert}`
Exemple 2 - les parents de Robert ou d'Alex (avoir Robert ou Alex comme enfant) : `a_enfant.{Robert, Alex}`
Exemple 3 - les parents de Robert et d'Alex (avoir Robert et Alex comme enfant) : `a_enfant.{Robert}`$\sqcap$`a_enfant.{Alex}`

3.5 Nomenclature

- $\mathcal{ALCI}$: $\mathcal{ALC}$ avec rôles inverses
- $\mathcal{ALCN}$: $\mathcal{ALC}$ avec restrictions de cardinalité non qualifiées
- $\mathcal{ALCIN}$: $\mathcal{ALC}$ avec rôles inverses et restrictions de cardinalité non qualifiées
- $\mathcal{ALCQ}$: $\mathcal{ALC}$ avec restrictions de cardinalité qualifiées
- $\mathcal{ALCIQ}$: $\mathcal{ALC}$ avec rôles inverses et restrictions de cardinalité qualifiées
- $\mathcal{ALCO}$: $\mathcal{ALC}$ avec concepts nominaux
- $\mathcal{ALCOI}$: $\mathcal{ALCO}$ avec rôles inverses
- $\mathcal{ALCON}$: $\mathcal{ALCO}$ avec restrictions de cardinalité non qualifiées
- $\mathcal{ALCOIN}$: $\mathcal{ALCO}$ avec rôles inverses et restrictions de cardinalité non qualifiées
- $\mathcal{ALCOQ}$: $\mathcal{ALCO}$ avec restrictions de cardinalité qualifiées
- $\mathcal{ALCOIQ}$: $\mathcal{ALCO}$ avec rôles inverses et restrictions de cardinalité qualifiées

4 Algorithme tableau pour $\mathcal{ALC}$

4.1 Règles de réécriture

Etant donné un ensemble de formules $\mathcal{F}$ écrites en $\mathcal{ALC}$ sous forme normale négative,

si $\mathcal{F}$ contient	ajouter à $\mathcal{F}$
$a : C \sqcap D$	$a : C$ et $a : D$
$a : \forall R.C$ et $\langle a, b \rangle : R$	$b : C$
$a : \exists R.C$	$\langle a, b \rangle : R$ et $b : C$, où b est un nouvel objet
$a : \neg A$ et $a : \neg A$	$a : \bot$
$a : C \sqcup D$	$a : C$ **ou** $a : D$

4.2 Principe de l'algorithme

L'algorithme consiste à développer l'arbre en appliquant récursivement les règles précédentes.
Une branche est dite fermée si elle contient une assertion du type $a : \bot$.
Une branche est dite complète si plus aucune règle ne s'appliqué.
L'algorithme s'applique si toutes les branches sont fermées ou si une branche est complète.
L'ensemble de formules $\mathcal{F}$ est insatisfiable si et seulement si toutes les branches de l'arbre sont fermées.