

Logiques de description, et ontologies en logiques de description

Toolbox 2 Modélisation des Systèmes – Mines Saint-Etienne

3 mai 2016

1 Ontologies en logique de description

Différentes logiques peuvent être utilisées pour la représentation de connaissances, la logique des prédicats en est une, et nous allons présenter les logiques de description.

1.1 Structures logiques de Tarski

(cette section est informative, elle ne fera pas l'objet d'une évaluation à la fin du cours.)

Nous présentons tout d'abord une structure logique très importante et très générale introduite par Jan Lukasiewicz et Alfred Tarski à la fin des années 1920. Une *structure logique* est définie comme une *paire* $\langle \mathcal{L}, Cn \rangle$, où \mathcal{L} représente le langage (i.e., les propositions exprimables dans la logique), et Cn est un opérateur de clôture défini par un certain nombre de règles. Cn associe à chaque ensemble de propositions X un autre ensemble de propositions $Cn(X)$, appelées conséquences de X , qui est obtenu en appliquant les règles en boucle (éventuellement infinie) sur les propositions de X . $Cn(X)$ est la clôture de X .

Tarski a proposé trois axiomes qui contraignent cet opérateur de sorte qu'il ait un comportement rationnel.

$$\begin{aligned} \forall X \subseteq \mathcal{L}, X \subseteq Cn(X) & \quad (\textit{inclusion}) \\ \forall X \subseteq \mathcal{L}, Cn(Cn(X)) = Cn(X) & \quad (\textit{itération}) \\ \forall X, Y \subseteq \mathcal{L}^2, X \subseteq Y \Rightarrow Cn(X) \subseteq Cn(Y) & \quad (\textit{monotonie}) \end{aligned}$$

De manière équivalente à l'opérateur de conséquence Cn , on peut définir une *relation d'inférence* entre les ensembles de connaissances :

$$X \vdash Y \Leftrightarrow Y \subseteq Cn(X)$$

On peut lire l'équation ci-dessus : X implique (en anglais : entails) Y si et seulement si Y est inclus dans l'ensemble des conséquences de X .

Ainsi, à partir des *connaissances* : un ensemble de propositions sous-ensemble de \mathcal{L} , on peut déduire d'autres propositions par un mécanisme d'inférence. On peut montrer que la relation d'inférence satisfait les propriétés suivantes :

$$\begin{aligned} \forall X, Y \subseteq \mathcal{L}^2, X \subseteq Y \Rightarrow Y \vdash X & \quad (\textit{reflexivité}) \\ \forall X, Y, Z \subseteq \mathcal{L}^3, X \vdash Y \text{ et } X \cup Y \vdash Z \Rightarrow X \vdash Z & \quad (\textit{transitivité}) \\ \forall X, Y, Z \subseteq \mathcal{L}^3, X \vdash Y \Rightarrow X \cup Z \vdash Y & \quad (\textit{affaiblissement}) \end{aligned}$$

1.2 Logiques de description, et ontologies en logiques de description

La syntaxe d'un langage de représentation des connaissances se caractérise par un ensemble de symboles et des règles de constructions permettant d'obtenir des formules à partir de ces symboles et de constructeurs spécifiques au langage. Les logiques de description (*Description Logics, ou DLs*) forment une famille de langages de représentation de connaissance. qui peuvent être utilisées pour représenter la connaissance terminologique d'un domaine d'application d'une façon structurée et formelle. Ces

langages ont été introduit dans les années 80, dans le but de rendre la représentation de connaissances plus *naturelle* qu'en logique du premier ordre. Ils ont néanmoins une sémantique formelle conforme au cadre des logiques de Tarski. Dans ce sens, nous pouvons dire que les DLs ont une sémantique descriptive formelle.

1.2.1 Syntaxe d'une logique de description

Une logique de description \mathcal{LD} divise la connaissance en deux parties :

- **T-box – les informations terminologiques** : un ensemble de formules relatives aux informations sur les notions basiques ou dérivées et comment elles sont reliées entre elles. Ces informations sont *génériques* ou *globales*, vraies dans tous les modèles et pour tous les individus.
- **A-box – les informations sur les individus** : un ensemble de formules relatives aux informations *spécifiques* ou *locales*, vraies pour certains individus particuliers.

Les logiques de description utilisent les notions de *concept*, de *rôle* et d'*individu* :

Définition 1 (Concept). Un concept d'une logique de description \mathcal{LD} correspond à une classe d'éléments et est interprété comme un ensemble dans un univers donné. Les concepts peuvent être atomiques (correspondant à un symbole de type concept), ou complexes obtenus en utilisant les constructeurs de concepts autorisés par le langage de \mathcal{LD} .

Définition 2 (Rôle). Un rôle d'une logique de description \mathcal{LD} correspond à un lien entre deux éléments et est interprété comme une relation binaire sur un univers donné. Les rôles peuvent être atomiques, ou complexes obtenus en utilisant les constructeurs de rôles autorisés par le langage de \mathcal{LD} .

Définition 3 (Individu). Un individu d'une logique de description \mathcal{LD} correspond à un élément d'un univers donné.

Définition 4 (Signature). Soit \mathcal{LD} une logique de description, $\mathcal{C} = \{C_1, C_2, \dots\}$ un ensemble fini de concepts atomiques, $\mathcal{R} = \{R_1, R_2, \dots\}$ un ensemble fini de rôles atomiques et $\mathcal{U} = \{a_1, a_2, \dots\}$ un ensemble fini d'individus. Pour $\mathcal{C}, \mathcal{R}, \mathcal{U}$ disjoints deux à deux, le triplet $\Sigma = \langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ est une signature de \mathcal{LD} .

Définition 5 (Formule). Soit \mathcal{LD} une logique de description, Une formule de \mathcal{LD} pour une signature Σ est l'un des axiomes autorisés par le langage de \mathcal{LD} .

Définition 6 (Ontologie). Soit \mathcal{LD} une logique de description, une ontologie de \mathcal{LD} est une paire $\langle \Sigma, F \rangle$ tel que Σ est une signature et $F = \langle T, A \rangle$ contient des axiomes T de la T-box, et des assertions A de la A-box.

Définition 7 (Element d'une ontologie). Un élément d'une ontologie $o = \langle \Sigma, F \rangle$ est soit une entité atomique de l'ontologie (e.g., un élément de la signature Σ), soit une entité complexe construite en utilisant des constructeurs du langage de l'ontologie. Dans la suite, on utilisera la notation $\mathcal{Q}_{\mathcal{LD}}(o)$ pour désigner l'ensemble des éléments d'une ontologie.

Le tableau 1 donne en exemple les syntaxes de chaque terme atomique, ainsi que différents constructeur de concept, constructeur de rôle, axiomes de la T-box et axiomes de la A-box. Le nom de la logique minimale qui les autorise est noté dans la colonne de droite.

Terme atomique	Syntaxe	Sémantique	
individu	a	$I(a) \in \Delta$	
concept atomique	A	$I(A) \subseteq \Delta$	
rôle atomique	R	$I(R) \subseteq \Delta \times \Delta$	(\mathcal{AL})
Constructeur de concepts	Syntaxe	Sémantique	nom
concept universel	\top	$I(\top) = \Delta$	(\mathcal{AL})
concept vide	\perp	$I(\perp) = \emptyset$	(\mathcal{AL})
conjonction	$C \sqcap D$	$I(C \sqcap D) = I(C) \cap I(D)$	(\mathcal{AL})
disjonction	$C \sqcup D$	$I(C \sqcup D) = I(C) \cup I(D)$	\mathcal{U}
négation	$\neg C$	$\Delta \setminus I(C)$	\mathcal{C}
restriction existentielle	$\exists R.C$	$\{x \mid \exists y, \langle x, y \rangle \in I(R) \wedge y \in I(C)\}$	(\mathcal{AL})
restriction de valeur	$\forall R.C$	$\{x \mid \forall y, \langle x, y \rangle \in I(R) \Rightarrow y \in I(C)\}$	(\mathcal{AL})
restriction numérique	$\leq nR$	$\{x \mid \text{card}(\{y, \langle x, y \rangle \in I(R)\}) \leq n\}$	\mathcal{N}
restriction numérique	$\geq nR$	$\{x \mid \text{card}(\{y, \langle x, y \rangle \in I(R)\}) \geq n\}$	\mathcal{N}
nominaux	$\{a_1, \dots, a_n\}$	$\{I(a_1), \dots, I(a_n)\}$	\mathcal{O}
Constructeur de rôles	Syntaxe	Sémantique	nom
conjonction de rôles	$R \sqcap S$	$I(R \sqcap S) = I(R) \cap I(S)$	$\cdot \sqcap$
disjonction de rôles	$R \sqcup S$	$I(R \sqcup S) = I(R) \cup I(S)$	$\cdot \sqcup$
complément de rôle	$\neg R$	$(\Delta \times \Delta) \setminus I(R)$	$\cdot \neg$
clôture transitive	R^+	<i>clôture transitive de</i> $I(R)$	$\cdot +$
rôle inverse	R^-	$\{\langle y, x \rangle \mid \langle x, y \rangle \in I(R)\}$	$\cdot ^-$
composition de rôles	$R \circ S$	$\{\langle x, z \rangle \mid \exists y, \langle x, y \rangle \in I(R) \wedge \langle y, z \rangle \in I(S)\}$	$\cdot \circ$
Axiomes de la T-box	Syntaxe	Contrainte d'interprétation	nom
subsomption	$C \sqsubseteq D$	$I(C) \subseteq I(D)$	(\mathcal{AL})
inclusion de rôles	$R \sqsubseteq S$	$I(R) \subseteq I(S)$	\mathcal{H}
transitivité de rôles	$\text{Trans}(R)$	$I(R) = I(R^+)$	\mathcal{S}
Axiomes de la A-box	Syntaxe	Contrainte d'interprétation	nom
appartenance à un concept	$C(a)$	$I(a) \in I(C)$	\mathcal{AL}
appartenance à un rôle	$R(a_1, a_2)$	$\langle I(a_1), I(a_2) \rangle \in I(R)$	\mathcal{AL}
identité	$a_1 = a_2$	$I(a_1) = I(a_2)$	

TABLE 1 – Syntaxe et sémantique de différents constructeurs et connecteurs de logiques de description. (Les colonnes 3-Sémantique et 4-nom de la logique minimale qui autorise cet axiome, sont données à titre d'information, et ne feront pas l'objet d'une évaluation à la fin du cours)

1.2.2 Sémantique des logiques de description

(cette section est informative, elle ne fera pas l'objet d'une évaluation à la fin du cours.)

La sémantique des logiques de description est définie comme suit :

Définition 8 (Interprétation). Soient \mathcal{LD} une logique de description et $\Sigma = \langle \mathcal{C}, \mathcal{R}, \mathcal{U} \rangle$ une signature de \mathcal{LD} . Une interprétation \mathcal{I} de Σ est une paire $\langle \Delta, I \rangle$, où le domaine d'interprétation Δ est un ensemble non-vide, et I est un triplet de fonctions : $I_{\mathcal{C}} : \mathcal{C} \rightarrow 2^{\Delta}$, $I_{\mathcal{R}} : \mathcal{R} \rightarrow 2^{\Delta \times \Delta}$, et $I_{\mathcal{U}} : \mathcal{U} \rightarrow \Delta$.

Définition 9 (Relation de satisfaction). Soit \mathcal{LD} une logique de description. A chaque axiome autorisé par le langage de \mathcal{LD} , est associé une contrainte d'interprétation. On dit qu'une interprétation \mathcal{I} satisfait une formule f si et seulement si la contrainte d'interprétation associée à l'axiome de f est satisfaite. On note cette relation $\mathcal{I} \models f$.

Le tableau 1 donne en exemple les relations de satisfaction de chaque terme atomique, constructeur de concept, constructeur de rôle, axiomes de la T-box et axiomes de la A-box.

Définition 10 (Modèle). Soient \mathcal{LD} une logique de description et $o = \langle \Sigma, F \rangle$ une ontologie de \mathcal{LD} . Une interprétation de Σ est un modèle de o si et seulement si elle satisfait l'ensemble des formules de F . On note cette relation $\mathcal{I} \models o$. On note également $\mathcal{M}(o)$ l'ensemble des modèles de o .

Les deux notions essentielles de raisonnement logique sont la consistance d'une ontologie et l'opérateur de conséquence sémantique (ou de manière équivalente, une relation d'inférence).

Définition 11 (Consistance). Soit \mathcal{LD} une logique de description, une ontologie o de \mathcal{LD} est consistante si et seulement si il existe au moins un modèle de o .

Une autre notion, celle de cohérence, est parfois présente dans la littérature, mais elle est associée à des sens différents. A priori, une ontologie sera dite incohérente si elle ne répond pas à certains critères que l'on peut associer à des règles de bon usage. Ces règles peuvent évidemment varier d'un groupe d'ingénieurs d'ontologie à un autre. Un critère classique consiste à dire qu'une ontologie est incohérente si une de ses classes est nécessairement vide d'individus, sous peine de rendre l'ontologie inconsistante.

Définition 12 (Conséquence sémantique, clôture). Soit \mathcal{LD} une logique de description, o une ontologie de \mathcal{LD} et f une formule de \mathcal{LD} . On dit que f est une conséquence sémantique de o lorsque tous les modèles de o satisfont f . Dans ce cas on note $o \models f$. L'ensemble des conséquences sémantiques de l'ontologie o est la clôture de o et est noté $Cn(o)$.

Définition 13 (Décidabilité). Soit \mathcal{LD} une logique de description, on dit que \mathcal{LD} est décidable si et seulement si pour toute ontologie o de \mathcal{LD} et pour toute formule f de \mathcal{LD} , il existe un algorithme permettant de déterminer si f est une conséquence sémantique de o ou non.

Définition 14 (Correction et complétude). Soit \mathcal{LD} une logique de description, o une ontologie de \mathcal{LD} et f une formule de \mathcal{LD} . Un algorithme A tel que $A(o, f) \in \{VRAI, FAUX\}$ est dit :

- *correct* si et seulement si pour toute ontologie o et pour toute formule f , $A(o, f) = VRAI \Rightarrow o \models f$;
- *complet* si et seulement si pour toute ontologie o et pour toute formule f , $o \models f \Rightarrow A(o, f) = VRAI$;

1.2.3 Langages de représentation des ontologies

Les ontologies sont exprimées à l'aide de langages formels tels que RDF, RDF Schema, et OWL 2, pour ne citer que les standards du World Wide Web Consortium (W3C).

Le Web Ontology Language (ou OWL 2) recommandé comme standard par le W3C, est basé sur les fondements théoriques des logiques de description. En ce sens, ce standard propose plusieurs sous-langages, chaque sous-langage correspondant à une logique de description bien connue, avec une certaine expressivité et une certaine complexité. Pour une application particulière, le choix du sous-langage à utiliser est conditionné par le rapport d'importance auquel on s'accorde entre l'expressivité du langage, et la complexité des calculs nécessaires pour raisonner sur l'ontologie. A la limite, le sous-langage qui autorise l'ensemble du langage proposé par OWL 2 est a une complexité telle qu'il est indécidable.

2 Quelques transformées en logique du premier ordre

Cette section précise la transformée entre les logiques de description et la logique du premier ordre pour quelques axiomes et constructeurs de concepts et de rôles.

2.1 Exemples de constructeurs de concepts

Concept universel - Thing

Syntaxe des logiques de description : \top

$$(\top)(x) \iff \top$$

Concept vide - Nothing

Syntaxe des logiques de description : \perp

$$(\perp)(x) \iff \perp$$

Conjonction - ObjectIntersectionOf(A B)

Syntaxe des logiques de description : $A \sqcap B$

$$(A \sqcap B)(x) \iff A(x) \wedge B(x)$$

Disjonction - ObjectUnionOf(A B)

Syntaxe des logiques de description : $A \sqcup B$

$$(A \sqcup B)(x) \iff A(x) \vee B(x)$$

Négation - ObjectComplementOf(A)

Syntaxe des logiques de description : $\neg A$

$$(\neg A)(x) \iff \neg A(x)$$

Enumération/nominaux - ObjectOneOf(A)

Syntaxe des logiques de description : $\{a_1, \dots, a_n\}$

$$(\{a_1, \dots, a_n\})(x) \iff x = a_1 \vee \dots \vee x = a_n$$

Restriction existentielle - ObjectSomeValuesFrom(f C)

Syntaxe des logiques de description : $\exists f.C$

$$(\exists f.C)(x) \iff (\exists y)[f(x, y) \wedge C(y)]$$

Restriction de valeur - ObjectAllValuesFrom(f C)

Syntaxe des logiques de description : $\forall f.C$

$$(\forall f.C)(x) \iff (\forall y)[f(x, y) \rightarrow C(y)]$$

Restriction à un individu - ObjectHasValue(f a)

Syntaxe des logiques de description : $\exists f.a$

$$(\exists f.a)(x) \iff f(x, a)$$

Existence - ObjectMinCardinality(1 f)

Syntaxe des logiques de description : $\geq 1.f$

$$(\geq 1.f)(x) \iff (\exists y)[f(x, y)]$$

Inexistence - ObjectMaxCardinality(0 f)

Syntaxe des logiques de description : $\leq 0.f$

$$(\leq 0.f)(x) \iff (\forall y)[\neg f(x, y)]$$

Fonctionnalité - ObjectMaxCardinality(1 f)

Syntaxe des logiques de description : $\leq 1.f$

$$(\leq 1.f)(x) \iff (\forall y, z)[f(x, y) \wedge f(x, z) \rightarrow y = z]$$

Classe réflexive - ObjectHasSelf(f)

Syntaxe des logiques de description : **HasSelf.f**

$$(\text{HasSelf}.f)(x) \iff f(x, x)$$

2.2 Exemples de constructeurs de rôles**Conjonction - ObjectPropertyIntersectionOf(f g)**

Syntaxe des logiques de description : $f \sqcap g$

$$(f \sqcap g)(x, y) \iff f(x, y) \wedge g(x, y)$$

Disjonction - ObjectPropertyUnionOf(f g)

Syntaxe des logiques de description : $f \sqcup g$

$$(f \sqcup g)(x, y) \iff f(x, y) \vee g(x, y)$$

Négation - ObjectPropertyComplementOf(f)

Syntaxe des logiques de description : $\neg f$

$$(\neg f)(f g) \iff \neg f(f g)$$

Inverse - ObjectPropertyInverseOf(f)

Syntaxe des logiques de description : f^-

$$(f^-)(x, y) \iff f(y, x)$$

2.3 Exemples d'axiomes de la T-box

Subsompion de concept - subClassOf(A B)

Syntaxe des logiques de description : $A \sqsubseteq B$

$$A \sqsubseteq B \iff (\forall x)[A(x) \rightarrow B(x)]$$

Equivalence de concept - equivalentClasses(A B)

Syntaxe des logiques de description : $A \equiv B$

$$A \equiv B \iff (\forall x)[A(x) \leftrightarrow B(x)]$$

Subsompion de rôle - subObjectPropertyOf(f g)

Syntaxe des logiques de description : $f \sqsubseteq g$

$$f \sqsubseteq g \iff (\forall x, y)[f(x, y) \rightarrow g(x, y)]$$

Domaine - objectPropertyDomain(f C)

Syntaxe des logiques de description : $(\geq 1.f) \sqsubseteq C$

$$(\geq 1.f) \sqsubseteq C \iff (\forall x)[(\exists y)[f(x, y)] \rightarrow C(x)]$$

Subsompion de rôle composé - PropertyChainAxiom(f₁ f₂ g)

Syntaxe des logiques de description : $f_1 \circ f_2 \sqsubseteq g$

$$f_1 \circ f_2 \sqsubseteq g \iff (\forall x, y, z)[f_1(x, y) \wedge f_2(y, z) \rightarrow g(x, z)]$$