Faculté des Sciences
Aix-Marseille Université

Année universitaire 2024/2025

Site : ☐ Luminy ☐ St-Charles ☒ St-Jérôme ☐ Cht-Gombert ☐ Aix-Montperrin ☐ Aubagne-SATIS
Sujet de : ☒ 1$^{\text{er}}$ semestre ☐ 2$^{\text{ème}}$ semestre ☐ Session 2 Durée de l'épreuve : 2h
Examen de : M2 Nom du diplôme : Master 2 Informatique Parcours SID
Code du module : SINCUB1 Libellé du module : Web sémantique, ontologie et raisonnement
Calculatrices autorisées : NON Documents autorisés : NON

1 Partie II : Logiques de description, ontologies et ASP

1.1 Exercice 1 (Logiques de description et ontologies : 9pts)

a) On se situe dans la logique de description de base $\mathcal{AL}$.

1. En considérant les deux concepts atomiques $Animal$ et $Raisonnable$ représentant respectivement la classe des animaux et celle de ceux qui raisonnent, donnez un axiome qui définit le concept $Personne$ représentant la classe des humains considérés comme étant des animaux qui raisonnent, puis donnez un autre axiome qui exprime le concepte $AnimalNonHumain$ représentant la classe des animaux autres que les humains.

2. En considérant les concepts définis en 1 et le rôle $vivantAvec$ où $vivantAvec(x, y)$ exprime le fait que x vit avec y, définissez le concept $Domestique$ exprimant la classe des animaux domestiques ne vivant qu'avec des humains, puis le concept $Sauvage$, représentant la classe des animaux non domestiques (ne vivant pas avec les humains).

3. En considérant, les concepts déjà définis, le concepte $Heureux$ représentant la classe des heureux, ainsi que le rôle $possede$ où $possede(x, y)$ exprime le fait que x possède y, définissez le concept $HeuProDom$ qui représente les heureux propriétaires d'animaux domestiques qui ne possèdent rien d'autre.

4. En considérant, les concepts déjà définis, le concept $Chat$ représentant les chats, ainsi que le rôle $aime$ où $aime(x, y)$ exprime le fait que x aime y, représentez le concept $PerChat$ des personnes qui n'aiment que les chats.

5. En considérant, les concepts déjà définis, exprimez le concept $DomChat$ des animaux domestiques qui n'aiment pas les chats.

6. En considérant, les concepts déjà définis, définissez le concept $PerPasChat$ des personnes qui n'aiment que les personnes qui n'aiment pas les chats.

b) Maintenant on se situe dans $\mathcal{AL}$ enrichie par certains constructeurs parmi $\{\mathcal{U}, \mathcal{E}, \mathcal{F}, \mathcal{I}, \mathcal{N}, \mathcal{Q}, \mathcal{C}\}$

1. Quels constructeurs faudrait-il rajouter à $\mathcal{AL}$ pour pouvoir exprimer le concept $SeulChat$ des personnes ne possédant rien d'autre qu'un seul chat. En considérant ces constructeurs, les concepts et les rôles de la partie (a), donnez alors l'axiome qui définit le concept $SeulChat$.

2. Que signifie l'axiome : $\top \sqsubseteq \leq 1\ aime$

3. Que signifie le concept $Toto$ de la $TBox$:
$Toto \equiv Personne \sqcap \exists possede.\top \sqcap \forall possede.Chat \sqcap \exists aime.Personne$

4. Que signifie le concept $Mistere$ de la $TBox$:
$\top \sqsubseteq \leq 1\ aime$
$\top \sqsubseteq \leq 1\ possede$
$Mistere \equiv Personne \sqcap \exists possede.\top \sqcap \forall possede.Chat \sqcap \exists aime.Personne$

Quelle est la différence entre le concept $Mister$ et le concept $Toto$ de la question précédente ?

1.2 Exercice 2 (Preuve dans les logiques de description : 6pts)

1. Soit la base de connaissances ($TBox$) suivante :
   $MonChien \equiv Chien \sqcap \forall aime.\forall aime.Chat \sqcap \forall aime.(Chien \sqcap \forall aime.\bot)$
   Après avoir donné au préalable la formule d'entrée de la méthode Tableaux obtenue en éliminant la $TBox$, prouvez à l'aide de cette même méthode (Tableaux), l'axiome de subsomption suivant :
   $MonChien \sqsubseteq \forall aime.\forall aime.Chien \sqcap Chien \sqcap \forall aime.Chien$

2. Soit la TBox exprimée par les axiomes suivants :

   $A \sqsubseteq B \sqcup C \sqcup D$
   $B \equiv \forall R.(E \sqcap F) \sqcap \geq 1R$
   $C \equiv \exists R.(E \sqcap \neg F)$
   $D \equiv \exists R.(F \sqcap \neg E)$

   Prouvez par la méthode Tableaux l'axiome de subsomption $A \sqsubseteq \exists R.(E \sqcup F)$

1.3 Exercice 3 (Answer Set Programming (ASP) : 5pts)

Soit le programme général : $\pi = \{a \leftarrow not\, b, b \leftarrow not\, c, c \leftarrow not\, d, d \leftarrow not\, e, e \leftarrow not\, f, f \leftarrow not\, a\}$

1. Calculez les réduits π^U, π^W π^X, π^Y, π^Z où $U = \emptyset$, $W = \{b\}$, $X = \{b, c\}$, $Y = \{b, d, f\}$ et $Z = \{c, d, e, f\}$. Dites si U, W, X, Y et Z sont des modèles stables de π.

2. Montrez (sans calculer les modèles stables) que π ne peut pas contenir de modèles stables avec zéro atomes, ni de modèles stables avec un seul atome, ni avec deux atomes, ni avec quatre et ni avec cinq ou six atomes.

3. Si l'on considère l'ordre suivant sur les six atomes de π : $\{a, b, c, d, e, f\}$, montrez que deux atomes consécutifs ne peuvent pas participer dans un même modèle stable de π.

4. Calculez les modèles stables de π.

2 Rappels : Logique de description et preuve par Tableaux

2.1 Logique de base $\mathcal{AL}$

En résumé, les descriptions possibles dans le langage $\mathcal{AL}$ sont les suivantes (on suppose que A est un concept atomique et C et que D sont des concepts atomiques ou complexes) :

A (concept atomique)
$\top$ (concept universel)
$\bot$ concept impossible)
$\neg A$ (négation atomique)
$C \sqcap D$ (intersection de concepts)
$\forall R.C$ (restriction de valeur)
$\exists R.\top >$ (quantification existentielle limitée)

2.2 Preuve dans les TBox avec axiomes de subsumtion ou d'inclusion $\sqsubseteq$

Une $TBox$ pourrait contenir des axiomes d'inclusion de la forme $C \sqsubseteq D$. Dans ce cas il faudrait faire un petit ajout au mécanisme d'inférence de la méthode Tableau qui consiste en :

- La transformation de l'axiome $C \sqsubseteq D$ en $\top \sqsubseteq \neg C \sqcup D$
- Dans la preuve par Tableaux, on pourrait alors ajouter le fait $(\neg C \sqcup D)(a)$ pour tout individu a.

Rappels

(1) Logiques de description, $\mathcal{AL}$ + Constructeurs

Définition 7 (Element d'une ontologie). Un élement d'une ontologie $o = \langle \Sigma, E \rangle$ 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 $Q_{\mathcal{LD}}(o)$ 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	$\bot$	$I(\bot) = \varnothing$	$(\mathcal{AL})$
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	$\neg C$	$\Delta \backslash I(C)$	$\mathcal{C}$
restriction existentielle	$\exists R.C$	$\{x \mid \exists y,\ \langle x, y\rangle \in I(R) \wedge y \in I(C)\}$	[redacted]
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	$\leq nR$	$\{x \mid card(\{y,\ \langle x, y\rangle \in I(R)\}) \leq n\}$	$\mathcal{N}$
restriction numérique	$\geq nR$	$\{x \mid card(\{y,\ \langle x, y\rangle \in I(R)\}) \geq n\}$	$\mathcal{N}$
nominaux	$\{a_1, \ldots, a_n\}$	$\{I(a_1), \ldots, I(a_n)\}$	$\mathcal{O}$
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	$\neg R$	$(\Delta \times \Delta) \backslash I(R)$	$\cdot^{\neg}$
clôture transitive	R^+	$clôturetransitivedeI(R)$	$\cdot^{+}$
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	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)

② Règles de la méthode de preuve par Tableau

Ainsi, pour faire une preuve, on crée d'abord l'énoncé dont on veut montrer l'insatisfaisabilité. Ensuite, en appliquant un séquence de règles, on ajoute de nouveaux énoncés. On crée ainsi ce qu'on appelle un *tableau*, qui est en fait un arbre, à cause des branchements qui peuvent y apparaître, comme dans l'exemple précédent.

Lorsqu'on a ajouté la contradiction $\square$ dans une branche, on dit qu'elle est fermée et on cesse d'en faire l'expansion. Lorsque toutes les branches d'un tableau sont fermées, on dit que le tableau est fermé et l'énoncé initial est considéré insatisfaisable. Si un tableau contient au moins une branche ouverte et qu'on ne réussit plus à appliquer des règles pour en faire l'expansion, le processus s'arrête et on conclut que l'énoncé initial n'est pas insatisfaisable.

6.2 Description formelle des règles

Soit un tableau et $\mathcal{A}$ un chemin complet qui va de la racine de l'arbre à une feuille. Le tableau 1 fournit la liste des règles d'expansion pour la logique $\mathcal{ALCN}$.

Règle	Condition	Action
règle-$\sqcap$	$\mathcal{A}$ contient $(C_1 \sqcap C_2)(x)$ et ne contient pas déjà les deux énoncés $C_1(x)$ et $C_2(x)$.	On ajoute à $\mathcal{A}$ les énoncés $C_1(x)$ et $C_2(x)$
règle-$\sqcup$	$\mathcal{A}$ contient $(C_1 \sqcup C_2)(x)$ et ne contient aucun des deux énoncés $C_1(x)$ et $C_2(x)$.	On ajoute à $\mathcal{A}$ le branchement suivant : $C_1(x)$ $C_2(x)$
règle-$\exists$	$\mathcal{A}$ contient $(\exists R.C)(x)$ et il n'existe aucun individu z tel que $R(x,z)$ et $C(z)$ sont aussi dans $\mathcal{A}$	On ajoute $R(x,y)$ et $C(y)$ à $\mathcal{A}$, où y est un nom d'individu qui n'existe pas déjà dans $\mathcal{A}$
règle-$\forall$	$\mathcal{A}$ contient $(\forall R.C)(x)$ et $R(x,y)$, mais ne contient pas $C(y)$	On ajoute $C(y)$ à $\mathcal{A}$
règle-$\geq$	$\mathcal{A}$ contient $(\geq n\, R)(x)$ et il n'y a pas dans $\mathcal{A}$ des individus $z_1, \ldots, z_n$ qui sont tous distincts (c'est-à-dire qu'on doit avoir explicitement dans $\mathcal{A}$ l'énoncé $z_i \neq z_j$ pour chaque paire possible avec cet ensemble d'individus) et qui sont tels que $\mathcal{A}$ contient la relation $R(x, z_i)$ pour tous ces individus $(1 \leq i \leq n)$.	Soit un ensemble de n individus dénotés par $y_1, \ldots, y_n$, qui sont des noms qui n'existent pas dans $\mathcal{A}$. On ajoute à $\mathcal{A}$ les énoncés $y_i \neq y_j$ pour chaque paire possible avec cet ensemble, ainsi que les énoncés $R(x, y_i)$ pour $(1 \leq i \leq n)$.
règle-$\leq$	$\mathcal{A}$ contient $(\leq n\, R)(x)$ et les énoncés $R(x,y_1), \ldots, R(x, y_{n+1})$. Il n'existe aucune identité $y_i = y_j$ dans $\mathcal{A}$ pour $(1 \leq i \leq n+1)$, $(1 \leq j \leq n+1), i \neq j$	Pour chaque paire possible (y_i, y_j) d'individus parmi y_i, y_{n+1}, on ajoute une nouvelle branche avec $y_i = y_j$.

TABLE 1 – Règles de tableau pour la logique $\mathcal{ALCN}$

Pour chaque règle appliquée, on ajoute les nouveaux énoncés à la fin à partir de la branche où on se trouve. On les ajoute généralement à la fin de tous les branchements à partir du point où on se trouve. ~~Supposons par exemple le tableau de la figure 2a, où le traitement de l'énoncé A entraîne l'ajout de l'énoncé B. Il faudra alors étendre le tableau de la manière illustrée à la figure 2b. Similairement, lorsqu'on doit ajouter un branchement dans un tableau, on l'ajoute à toutes~~