

**Durée 2h - aucun document autorisé**

Le barème n'est donné qu'à titre indicatif

**1 Logique propositionnelle**

**Exercice 1** – Méthode des tableaux sémantiques – 2 points

Démontrer la validité de  $F = ((p \wedge q) \rightarrow r) \rightarrow ((p \rightarrow r) \vee (q \rightarrow r))$ , c'est-à-dire  $\models F$ , avec la méthode des tableaux sémantiques, en indiquant à chaque étape la formule traitée et la règle appliquée.

**Correction :** Il faut démontrer l'insatisfiabilité de  $\neg F = \neg((p \wedge q) \rightarrow r \rightarrow ((p \rightarrow r) \vee (q \rightarrow r)))$

Avec la règle  $\alpha$  sur  $\neg(\phi_1 \rightarrow \phi_2)$  on obtient :

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$

On peut maintenant appliquer la règle  $\alpha$  sur  $\neg(\phi_1 \vee \phi_2)$  ce qui donne :

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$
$\neg(p \rightarrow r)$
$\neg(q \rightarrow r)$

d'où, avec la règle  $\alpha$  sur  $\neg(\phi_1 \rightarrow \phi_2)$  que l'on applique deux fois :

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$
$\neg(p \rightarrow r)$
$\neg(q \rightarrow r)$
$p$
$q$
$\neg r$

En appliquant la règle  $\beta$   $\phi_1 \rightarrow \phi_2$  on obtient deux tableaux :

$T_1 :$

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$
$\neg(p \rightarrow r)$
$\neg(q \rightarrow r)$
$p$
$q$
$\neg r$
$\neg(p \wedge q)$

et  $T_2 :$

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$
$\neg(p \rightarrow r)$
$\neg(q \rightarrow r)$
$p$
$q$
$\neg r$
$r$

qui contient un clash.

L'application de la règle  $\beta$   $\neg(\phi_1 \wedge \phi_2)$  sur  $T_1$  donne encore deux tableaux :

$T_{1,1} :$

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$
$\neg(p \rightarrow r)$
$\neg(q \rightarrow r)$
$p$
$q$
$\neg r$
$\neg(p \wedge q)$
$\neg p$

et  $T_{1,2} :$

$(p \wedge q) \rightarrow r$
$\neg((p \rightarrow r) \vee (q \rightarrow r))$
$\neg(p \rightarrow r)$
$\neg(q \rightarrow r)$
$p$
$q$
$\neg r$
$\neg(p \wedge q)$
$\neg q$

qui contiennent tous les deux un clash.

**Exercice 2** – Système de Hilbert – 2 points

On pose  $G = ((P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R)))$ . Démontrer que  $\vdash G$  avec le système de Hilbert.

**Correction :**

1.  $(P \rightarrow Q)$  Hypothèse 1
2.  $(Q \rightarrow R)$  Hypothèse 2
3.  $P$  Hypothèse 3
4.  $Q$  Modus Ponens 1, 3
5.  $R$  Modus Ponens 4, 2

d'où  $(P \rightarrow Q), (Q \rightarrow R), P \vdash R$

en appliquant trois fois le théorème de la déduction, on obtient

1.  $(P \rightarrow Q), (Q \rightarrow R) \vdash (P \rightarrow R)$  puis
2.  $(P \rightarrow Q) \vdash ((Q \rightarrow R) \rightarrow (P \rightarrow R))$  et enfin
3.  $\vdash ((P \rightarrow Q) \rightarrow ((Q \rightarrow R) \rightarrow (P \rightarrow R)))$

## 2 Logique des prédicats du premier ordre

**Exercice 3** – Représentation en logique des prédicats du premier ordre – 3,5 points

Un plateau de jeu est composé de  $3 \times 3$  cases. Sur ce plateau, des cubes peuvent être empilés, jusqu'à dix cubes, les uns sur les autres.

Les prédicats  $noir(x, y, z)$  et  $blanc(x, y, z)$  indiquent respectivement qu'un cube noir (resp. blanc) se trouve sur la case correspondant à la ligne  $x$  et la colonne  $y$ , et à l'étage  $z$  (on parle de cube à la position  $(x, y, z)$ ). Les domaines de définition sont donc  $D = \{1, 2, 3\}$  pour  $x$  et  $y$ , et  $D' = \{1, \dots, 10\}$  pour  $z$ . Par convention un cube posé directement sur le plateau est au niveau 1. La fonction  $dessous(z)$  permet de renvoyer l'étage en dessous de l'étage  $z$ , et par convention  $dessous(1) = 1$ .

Par exemple,  $noir(1, 2, dessous(3))$  est vrai s'il y a un cube noir à la ligne 1, colonne 2, et à l'étage 2.

Exprimez sous forme de logique propositionnelle les énoncés suivants :

1. La partie est commencée, *ie.* il y a au moins un cube posé sur le plateau.

**Correction :**  $\exists x \exists y (noir(x, y, 1) \vee blanc(x, y, 1))$

2. Aucune position ne comporte à la fois un cube noir et un cube blanc

**Correction :**  $\forall x \forall y \forall z (\neg noir(x, y, z) \vee \neg blanc(x, y, z))$

3. Il existe une ligne posée sur le plateau où tous les cubes sont noirs

**Correction :**  $\exists x \forall y noir(x, y, 1)$

4. Il existe une tour de 10 cubes où tous les cubes sont de la même couleur

**Correction :**  $\exists x \exists y (\forall z noir(x, y, z) \vee \forall z blanc(x, y, z))$

5. Aucun cube ne tient magiquement en l'air, *ie.* si un cube est à l'étage  $z$  il y a forcément un autre cube juste en dessous.

**Correction :**  $\forall x \forall y \forall z (noir(x, y, z) \vee blanc(x, y, z)) \rightarrow (noir(x, y, dessous(z)) \vee blanc(x, y, dessous(z)))$

6. En regardant du ciel, tous les cubes que l'on voit sont noirs, *ie.* tous les cubes au sommet des tours sont noirs.

**Correction :**

$\forall x \forall y ((\neg noir(x, y, 1) \wedge \neg blanc(x, y, 1)) \vee \exists z noir(x, y, dessous(z)) \wedge \neg (noir(x, y, z) \vee blanc(x, y, z)))$

**Exercice 4** – Règle de résolution – 3 points

Soit un ensemble de clauses

$$S = \{ \begin{array}{l} \neg P(x) \vee Q(f(x), x), \\ \neg P(x) \vee \neg Q(y, x) \vee R(y), \\ \neg R(y) \vee \neg S(y) \vee \neg Q(y, x), \\ \neg S(y) \vee R(y), \\ P(a), \\ \neg R(x) \vee S(x) \end{array} \}$$

où  $P$ ,  $Q$ ,  $R$  et  $S$  sont des prédicats,  $x$  et  $y$  sont des variables,  $a$  est une constante et  $f$  une fonction unaire.

Démontrer l'insatisfiabilité de  $S$  avec la règle de résolution en adoptant une stratégie linéaire.

**Correction :**

1.  $\neg P(x) \vee Q(f(x), x)$
2.  $\neg P(x) \vee \neg Q(y, x) \vee R(y)$
3.  $\neg R(y) \vee \neg S(y) \vee \neg Q(y, x)$
4.  $\neg S(y) \vee R(y)$
5.  $P(a)$
6.  $\neg R(x) \vee S(x)$
7.  $Q(f(a), a)$  résolution 5, 1
8.  $\neg P(a) \vee R(f(a))$  résolution 7, 2
9.  $R(f(a))$  résolution 8, 5
10.  $\neg S(f(a))$  résolution 9, 4
11.  $\neg R(f(a))$  résolution 10, 6
12.  $\square$  résolution 10, 9 C.Q.F.D.

### 3 Programmation logique

**Exercice 5** – Programmation du prédicat `elimination` – 1,5 points

Définir, en Prolog, le prédicat `elimination(X, L, R)` qui instancie la liste `R` avec une liste qui comprend tous les éléments de `L` distincts de `X`.

**Correction :**

```
elimination(_, [], []).
elimination(X, [X|Q], R) :- elimination(X, Q, R).
elimination(X, [T|Q], [T|R] :- X \= T, elimination(X, Q, R).
```

**Exercice 6** – Programmation de la différence de deux listes – 1,5 points

Définir, en Prolog, le prédicat `différence(L, LL, I)` qui instancie la liste `I` avec une liste qui comprend tous les éléments appartenant à `L` qui n'appartiennent pas à `LL` en faisant appel au seul prédicat `elimination`.

**Correction :**

```
différence(L, [], L).
différence(L, [X|Q], LL) :- elimination(X, L, P), différence(P, Q, LL).
```

### 4 Logiques de description

**Exercice 7** – Représentation en logique de description  $\mathcal{ALC}$  – 2 points

Représenter les propositions 1, 2, 3, 4, 5, 6 en logique de description  $\mathcal{ALC}$  en faisant appel aux concepts *Fleur*, *Dichlamydée*, *Monochlamydée*, *Achlamydée*, *Calice*, *Corolle*, *Euphorbe* et *Ortie* et au rôle *possède*.

1. Les dichlamydées sont des fleurs qui possèdent à la fois un calice et une corolle
2. Les monochlamydées sont des fleurs ne possédant qu'un calice ou qu'une corolle
3. Les achlamydées sont des fleurs qui n'ont ni calice, ni corolle
4. Les euphorbes sont des achlamydées
5. Les orties sont des monochlamydées
6. Les euphorbes n'ont pas de corolle

**Correction :**

1. Les dichlamydées sont des fleurs qui possèdent à la fois un calice et une corolle  $Dichlamydées \sqsubseteq Fleur \sqcap \exists \text{possède.Calice} \sqcap \exists \text{possède.Corolle}$
2. Les monochlamydées sont des fleurs ne possédant qu'un calice ou qu'une corolle  $Monochlamydées \sqsubseteq Fleur \sqcap (\exists \text{possède.Calice} \sqcup \exists \text{possède.Corolle}) \sqcap \neg(\exists \text{possède.Calice} \sqcap \exists \text{possède.Corolle})$
3. Les achlamydées n'ont ni calice, ni corolle  $Achlamydées \sqsubseteq Fleur \sqcap \neg(\exists \text{possède.Calice}) \sqcap \neg(\exists \text{possède.Corolle})$
4. Les euphorbes sont des achlamydées  $Euphorbes \sqsubseteq Achlamydées$
5. Les orties sont des monochlamydées  $Orties \sqsubseteq Monochlamydées$
6. Les euphorbes n'ont pas de corolle  $Euphorbes \sqsubseteq \neg \exists \text{possède.Corolle}$

**Exercice 8 – Démonstration en logique de description  $\mathcal{ALC}$  – 2 points**

Démontrer avec la méthode des tableaux sémantiques que si 1, 2, 3, 4, et 5 sont vrais, alors 6 est vrai, autrement dit que  $1, 2, 3, 4, 5 \models 6$ .

N.B. Chaque concept et rôle peut être abrégé en ses deux premières lettres pour alléger.

**Correction :** Il suffit de démontrer que  $1, 2, 3, 4, 5, \neg 6 \vdash \perp$

Pour cela, tout d'abord on réécrit  $\neg 6$ , c'est-à-dire  $\neg(Euphorbes \sqsubseteq \neg \exists \text{possède.Corolle})$  en  $Euphorbes \sqcap \exists \text{possède.Corolle}$  ce qui donne, par application de  $R_{\sqcap}$ ,  $Euphorbes$  et  $\exists \text{possède.Corolle}$

Ensuite, on considère 4 que l'on réécrit en  $\neg Euphorbes \sqcup Dichlamydées$ . Avec  $R_{\sqcup}$ , cela donne deux tableaux, l'un avec  $\neg Euphorbes$  qui est contradictoire. Le second avec  $Achlamydées$

On réécrit ensuite  $Achlamydées \sqsubseteq Fleur \sqcap \neg(\exists \text{possède.Calice}) \sqcap \neg(\exists \text{possède.Corolle})$  en  $\neg Achlamydées \sqcup (Fleur \sqcap \neg(\exists \text{possède.Calice}) \sqcap \neg(\exists \text{possède.Corolle}))$

Par  $R_{\sqcup}$ , cela donne deux tableaux dont un contradictoire et l'autre qui comprend :  $(Fleur \sqcap \neg(\exists \text{possède.Calice}) \sqcap \neg(\exists \text{possède.Corolle}))$

En appliquant  $R_{\sqcap}$  sur ce dernier, on obtient :  $Fleur, \neg(\exists \text{possède.Calice}), \neg(\exists \text{possède.Corolle})$  ce qui donne une contradiction avec  $\exists \text{possède.Corolle}$ .

## 5 Logique modale

**Exercice 9 — 2,5 points**

On considère la logique modale  $K$ .

1. Montrer que la formule  $F_1 = (a \rightarrow \Diamond a) \rightarrow (\Box a \rightarrow a)$  n'est pas valide dans cette logique.
2. Montrer par raisonnement sémantique que la formule  $F_2 = (a \rightarrow \Diamond a) \rightarrow (\Box \neg a \rightarrow \neg a)$  est valide dans cette logique.

**Correction :**

1. Il suffit de montrer un contre-exemple (un modèle et un monde) où la formule n'est pas satisfaite. Par exemple un monde unique, sans relation d'accessibilité, où  $a$  est faux, suffit.
2. Par raisonnement sémantique. Supposons par l'absurde qu'il existe un modèle  $M$  et un monde  $w$  tel que  $M, w \not\models (a \rightarrow \Diamond a) \rightarrow (\Box \neg a \rightarrow \neg a)$ . Alors on doit avoir (i)  $M, w \models a \rightarrow \Diamond a$  et (ii)  $M, w \not\models \Box \neg a \rightarrow \neg a$ . Et donc suivant (ii)  $M, w \models \Box \neg a$  et  $M, w \not\models \neg a$ . Mais alors on a  $M, w \models a$ , et par (i)  $M, w \models \Diamond a$ . Or  $M, w \models \Box \neg a$ , ce qui constitue une contradiction.

## 6 Annexe

### 6.1 Système de Hilbert

#### Rappel : le système de Hilbert pour la logique des prédicats

Le système de Hilbert est caractérisé par trois schémas d'axiomes et une règle d'inférence :

- Schémas d'axiomes :
    - SA1** :  $A \rightarrow (B \rightarrow A)$
    - SA2** :  $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$
    - SA3** :  $(\neg B \rightarrow \neg A) \rightarrow (A \rightarrow B)$
    - SA4 (règle  $\forall$ )** :  $\forall x A(x) \rightarrow A(r)$
  - Règles d'inférence
    - Modus Ponens** :  $A, A \rightarrow B \vdash B$
    - Généralisation** :  $(C \rightarrow A(t)) \vdash (C \rightarrow \forall x A(x))$
  - La déduction d'une formule  $A$  dans une théorie  $\Delta$  est une suite finie  $A_0, \dots, A_n$  telle que  $A_n = A$  et pour tout  $i$ ,
    - $A_i$  est l'instanciation de l'un des axiomes,
    - $A_i$  est l'une des hypothèses, c'est-à-dire  $A_i \in \Delta$
    - $A_i$  est obtenue par l'application d'une règle d'inférence, Modus Ponens sur  $A_j$  et  $A_k$  ou Généralisation sur  $A_j$ , avec  $j < i$  et  $k < i$ .
- On peut aussi appliquer toutes les substitutions nécessaires, à condition de les effectuer dans l'ensemble de la formule.
- Si on trouve une telle suite, on peut noter  $\Delta \vdash A$

On peut de plus utiliser le théorème de la déduction :

$$A_1, A_2, \dots, A_n \vdash B \quad \text{si et seulement si} \quad A_1, A_2, \dots, A_{n-1} \vdash (A_n \rightarrow B)$$

### 6.2 Méthode des tableaux sémantiques

La méthode des tableaux sémantiques permet d'établir si un ensemble de fomules logiques est valide, satisfiable ou insatisfiable.

#### 6.2.1 Composantes

La méthode des tableaux est basée sur des règles syntaxiques de décomposition, qui distinguent deux types de formules, nommés  $\alpha$  et  $\beta$ .

Formule $\alpha$	$\alpha_1$	$\alpha_2$	Formule $\beta$	$\beta_1$	$\beta_2$
$\neg\neg\varphi$	$\varphi$	$\varphi$	$\varphi_1 \vee \varphi_2$	$\varphi_1$	$\varphi_2$
$\varphi_1 \wedge \varphi_2$	$\varphi_1$	$\varphi_2$	$\neg(\varphi_1 \wedge \varphi_2)$	$\neg\varphi_1$	$\neg\varphi_2$
$\neg(\varphi_1 \vee \varphi_2)$	$\neg\varphi_1$	$\neg\varphi_2$	$\varphi_1 \rightarrow \varphi_2$	$\neg\varphi_1$	$\varphi_2$
$\neg(\varphi_1 \rightarrow \varphi_2)$	$\varphi_1$	$\neg\varphi_2$	$\neg(\varphi_1 \leftrightarrow \varphi_2)$	$\neg(\varphi_1 \rightarrow \varphi_2)$	$\neg(\varphi_2 \rightarrow \varphi_1)$
$\varphi_1 \leftrightarrow \varphi_2$	$\varphi_1 \rightarrow \varphi_2$	$\varphi_2 \rightarrow \varphi_1$			

#### 6.2.2 Satisfiabilité

La recherche d'un modèle pour un ensemble de formules  $\mathcal{F}$  par la méthode des tableaux peut être représentée de différentes façons, nous utilisons ici une forme arborescente.

- Initialisation : créer un nœud racine, étiqueté par l'ensemble  $\mathcal{F}$  et marqué comme non traité
- Décomposition itérative : choisir un nœud non traité et le marquer comme traité

- si l'étiquette du nœud contient deux littéraux complémentaires, marquer le nœud comme fermé
- sinon, si toutes les formules associées au nœud sont des variables propositionnelles, marquer le nœud comme ouvert
- sinon, choisir une formule  $F$  de l'étiquette du nœud
  - si elle est de type  $\alpha$ 
    - créer un sous-nœud marqué comme non traité
    - lui associer l'étiquette  $\mathcal{F} \setminus \{F\} \cup \{\alpha_1, \alpha_2\}$  où  $\alpha_1$  et  $\alpha_2$  sont les formules obtenues par réécriture de  $F$

Si l'arbre contient une feuille ouverte, alors  $\mathcal{F}$  est satisfiable.

Si toutes les feuilles de l'arbre sont fermées, alors  $\mathcal{F}$  est insatisfiable.

## 6.3 Logique de description $\mathcal{ALC}$

### 6.3.1 Alphabet

- Concepts atomiques :  $A, B, C, \dots$
- Rôles atomiques :  $r, s, u, v, \dots$
- Symboles :  $\{\sqcup, \sqcap, \exists, \forall, \neg, \top, \perp, \cdot\}$
- Instances de concepts :  $a, b, \dots$

### 6.3.2 Grammaire

concept ::=  $\langle$ concept atomique $\rangle$   
 $| \top$   
 $| \perp$   
 $| \neg$  $\langle$ concept $\rangle$   
 $| \langle$ concept $\rangle \sqcap \langle$ concept $\rangle$   
 $| \langle$ concept $\rangle \sqcup \langle$ concept $\rangle$   
 $| \exists \langle$ rôle $\rangle. \langle$ concept $\rangle$   
 $| \forall \langle$ rôle $\rangle. \langle$ concept $\rangle$

role ::=  $\langle$ role atomique $\rangle$   
 $| \langle$ role $\rangle \sqcap \langle$ role $\rangle$   
 $| \langle$ role $\rangle \sqcup \langle$ role $\rangle$

#### Rappels : règles pour mettre en œuvre la méthode des tableaux dans $\mathcal{ALC}$

On met d'abord les formules sous *forme normale négative*. Cela veut dire que l'on remplace les subsumptions du type  $A \sqsubseteq B$  par  $\neg A \sqcup B$  puis que l'on "rentre" les négations à l'intérieur des formules en changeant  $\neg(A \sqcup B)$  en  $\neg A \sqcap \neg B$ ,  $\neg(A \sqcap B)$  en  $\neg A \sqcup \neg B$ ,  $\neg\neg A$  en  $A$ ,  $\neg\forall r.C$  en  $\exists r.\neg C$  et  $\neg\exists r.C$  en  $\forall r.\neg C$ .

Il y a ensuite cinq règles à appliquer sur les tableaux  $\mathcal{A}$  issus des *ABox* :

- $R_{\sqcap}$  : si  $P \sqcap Q \in \mathcal{A}$  et soit  $P \notin \mathcal{A}$  soit  $Q \notin \mathcal{A}$ , alors ajouter  $\mathcal{A}' = \mathcal{A} \cup \{P, Q\}$  comme fils de  $\mathcal{A}$
- $R_{\sqcup}$  : si  $P \sqcup Q \in \mathcal{A}$  et ni  $P \in \mathcal{A}$  ni  $Q \in \mathcal{A}$ , alors ajouter les tableaux  $\mathcal{A}' = \mathcal{A} \cup \{P\}$  et  $\mathcal{A}'' = \mathcal{A} \cup \{Q\}$  comme fils de  $\mathcal{A}$
- $R_{\exists}$  : si  $\exists r.C \in \mathcal{A}$  et s'il n'existe pas de constante  $z$  telle que  $\langle x, z \rangle : r \in \mathcal{A}$  et  $z : C \in \mathcal{A}$ , alors ajouter le tableau  $\mathcal{A}' = \mathcal{A} \cup \{\langle x, z \rangle : r, z : C\}$  comme fils de  $\mathcal{A}$
- $R_{\forall}$  : si  $\forall r.C \in \mathcal{A}$ ,  $\langle x, y \rangle : r \in \mathcal{A}$  et  $y : C \notin \mathcal{A}$ , alors ajouter le tableau  $\mathcal{A}' = \mathcal{A} \cup \{y : C\}$  comme fils de  $\mathcal{A}$

Enfin, notons que s'il y a une *TBox*  $\mathcal{T}$  qui contient une formule  $C$ , on peut ajouter pour toute constante  $a$  l'assertion  $a : C$  dans la *ABox*  $\mathcal{A}$ .