Outils Logiques - 2018-2019, Examen Session 2

Consignes *Durée : 3h. L'utilisation de tout document ou dispositif électronique est* **interdite**. *Toute réponse doit être* **justifiée**. *Le barème est donné à titre* **indicatif**.

Exercice 1 (algèbre initiale et terminaison) [3,5 pnt] *Soit* $A^* = \{a, b, c\}^*$ *l'ensemble des mots finis sur l'alphabet* $\{a, b, c\}$.

1. *Définissez une signature* Σ *telle que les éléments de l'algèbre initiale* T_Σ *sont en correspondance bijective avec les éléments de l'ensemble* A^*.
2. *Soit* $\mathbf{N}$ *l'ensemble des nombres naturels. Définissez une* Σ*-algèbre* $\underline{A}$ *sur* $\mathbf{N}$ *telle que le morphisme (unique)* $h : T_\Sigma \to \underline{A}$ *correspond à la fonction qui compte le nombre de caractères dans un mot fini.*

Soit $\to$ *une relation binaire sur* A^* *telle que :* $w \to w'$ *ssi on peut trouver* w_1, w_2 *tels que* ($w = w_1 aa w_2$ *et* $w' = w_1 bc w_2$) *ou* ($w = w_1 bb w_2$ *et* $w' = w_1 ac w_2$) *ou* ($w = w_1 ccc w_2$ *et* $w' = w_1 ac w_2$). *Donc* w *se réduit à* w' *ssi* w' *est obtenu de* w *en remplaçant ou bien un sous-mot* aa *par* bc, *ou bien un sous-mot* bb *par* ac *ou bien un sous-mot* ccc *par* ac.

3. *La relation* $\to$ *est-elle transitive ?*
4. *La relation* $\to$ *termine-t-elle ?*

Exercice 2 (définissabilité et calcul des séquents) [7,5 pnt] *On rappelle que* $\mathbf{2} = \{0, 1\}$, *la fonction unaire* $NOT : \mathbf{2} \to \mathbf{2}$ *satisfait* $NOT(0) = 1$ *et* $NOT(1) = 0$ *et la fonction binaire* $XOR : \mathbf{2}^2 \to \mathbf{2}$ *satisfait* $XOR(0,0) = XOR(1,1) = 0$ *et* $XOR(1,0) = XOR(0,1) = 1$.

1. *Construisez une formule* A *en forme normale disjonctive (DNF) et une formule* B *en forme normale conjonctive (CNF) telles que* $var(A) = var(B) = \{x, y\}$, A *définit la fonction XOR et* B *définit la fonction XOR aussi.*
2. *Soit* $SSI : \mathbf{2}^2 \to \mathbf{2}$ *la fonction telle que* $SSI(x, y) = NOT(XOR(x, y))$. *Quelles sont les fonctions unaires* $f : \mathbf{2} \to \mathbf{2}$ *qui sont définissables par composition des fonctions XOR et SSI ?*
3. *Il n'est pas possible de définir toutes les fonctions binaires* $g : \mathbf{2}^2 \to \mathbf{2}$ *par composition des fonctions XOR et SSI. Proposez au moins un exemple d'une telle fonction. Pouvez-vous prouver que la fonction dans votre exemple n'est pas définissable ? Suggestion : il peut être utile de considérer les polynômes sur* $\mathbf{Z}_2$ *(les entiers modulo 2) définissables avec les fonctions XOR et SSI.*

On rappelle que l'opérateur logique $\oplus$ *désigne le* ou exclusif *et qu'il est donc interprété par la fonction XOR. On rappelle aussi qu'une règle d'inférence dans le calcul des séquents a la forme :*

$$\frac{\Gamma_1 \vdash \Delta_1 \cdots \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta}$$

Une règle est valide *si la validité des séquents* $\Gamma_i \vdash \Delta_i$ *pour* $i = 1, \ldots, n$ *(les hypothèses) implique la validité du séquent* $\Gamma \vdash \Delta$ *(la conclusion) et elle est* réversible *si la validité du séquent* $\Gamma \vdash \Delta$ *implique la validité des séquents* $\Gamma_i \vdash \Delta$ *pour* $i = 1, \ldots, n$.

4. *Rappelez l'axiome et les règles pour la négation dont on dispose dans le calcul des séquents.*
5. *Donnez un exemple de règle d'inférence valide mais pas réversible dont la conclusion a la forme* $\Gamma \vdash (A \oplus B), \Delta$.

6. *Proposez une règle d'inférence valide et réversible dont la conclusion a la forme* $\Gamma \vdash (A \oplus B), \Delta$ *(la règle traite donc le* $\oplus$ *à droite de* $\vdash$*).*
7. *Proposez une règle d'inférence valide et réversible dont la conclusion a la forme* $\Gamma, (A \oplus B) \vdash \Delta$ *(la règle traite donc le* $\oplus$ *à gauche de* $\vdash$*).*
8. *Utilisez les règles des points 4, 6 et 7 pour construire une preuve du séquent* $\vdash (A \oplus \neg A)$.
9. *Utilisez les règles des points 4, 6 et 7 pour construire une preuve du séquent* $(A \oplus A) \vdash$.

Exercice 3 (réfutation et résolution) [3 pnt] *Dans cet exercice, on utilise la notation ensembliste pour les formes normales conjonctives (CNF). En particulier, une CNF est un ensemble de clauses et une clause est un ensemble de littéraux. On dit qu'une clause est* positive *si elle contient seulement des littéraux positifs, c'est-à-dire sans négation (notez qu'une clause vide est une clause positive). Montrez ou donnez un contre-exemple aux assertions suivantes.*

1. *Si* A *est une CNF réfutable alors elle doit contenir au moins une clause positive.*
2. *Si* A *est une CNF satisfaisable alors elle contient seulement des clauses positives.*
3. *Pour toute CNF* A *qui ne contient pas la clause vide on peut trouver une affectation qui satisfait toutes les clauses positives contenues en* A.

Exercice 4 (CNF et BDD) [6,5 pnt] *On considère la contrainte :*

$$\Sigma_{i=1,\ldots,n} x_i = (x_1 + \cdots + x_n) = 1 \tag{1}$$

où $n \geq 2$ *et* $x_i \in \mathbf{2} = \{0, 1\}$. *NB Ici* Σ *et* $+$ *dénotent la somme sur les entiers. La contrainte dit donc qu'exactement un* x_i *vaut* 1 *et tous les autres valent* 0.

1. *Pour chaque* $n \geq 2$, *trouvez une formule en forme normale conjonctive (CNF)* A_n *telle que* $var(A_n) = \{x_1, \ldots, x_n\}$ *et une affectation* v *satisfait* A_n *ssi* $\Sigma_{i=1,\ldots,n} v(x_i) = 1$.
2. *Pour* $n = 4$, *donnez le schéma d'un BDD réduit qui correspond à la contrainte (1), à savoir un BDD qui calcule la fonction* $f : \mathbf{2}^4 \to \mathbf{2}$ *telle que* $f(x_1, \ldots, x_4) = 1$ *si* $\Sigma_{i=1,\ldots,4} x_i = 1$ *et* $f(x_1, \ldots, x_4) = 0$ *autrement.*
3. *On suppose maintenant* $n \geq 2$. *Combien de noeuds contient exactement le BDD réduit ? Ce nombre depend-t-il de l'ordre choisi pour les variables ?*
4. *On suppose qu'une 3-CNF est une CNF dont toutes les clauses contiennent exactement* 3 *littéraux. Pour* $n \geq 2$ *soit* A_n *la CNF du point 1. Est-ce possible de construire une 3-CNF* B_n *qui est satisfaisable ssi* A_n *est satisfaisable ? (*B_n *peut contenir plus de variables que* A_n*).*
5. *Une H2-CNF est une CNF dont chaque clause satisfait au moins une des conditions suivantes : (i) elle contient au plus 2 littéraux, (ii) elle contient au plus un littéral positif (c'est-à-dire sans négation). Soit* B *une 3-CNF. Est-ce possible de construire une H2-CNF* B' *qui est satisfaisable ssi* B *est satisfaisable ? (*B' *peut contenir plus de variables que* B*).*
6. *Votre collègue pense avoir un algorithme polynomial en temps pour déterminer si une formule H2-CNF est satisfaisable. Faut-il le croire ?*

Solution à l'exercice 1

1. [0,5 pnt] On prend $\Sigma = \{\epsilon^0, a^1, b^1, c^1\}$ on a donc un symbole de constante et trois symboles de fonctions unaires.
2. [0,5 pnt] On prend $\epsilon^A = 0 \in \mathbf{N}$ et $a^A = b^A = c^A = (x \mapsto x+1)$, la fonction successeur.
3. [0,5 pnt] Non, par exemple : $baa \to bbc \to cac$ mais $baa \not\to cac$.
4. [2 pnt] Oui. Par exemple, on peut remarquer que : (i) la troisième règle diminue le nombre de caractères et (ii) la première et la deuxième règle n'augmentent pas le nombre de caractères et diminuent le nombre de caractères a ou b.

   Donc si $|w|$ est le nombre de caractères dans w et $|w|_{ab}$ est le nombre de caractères dans w qui sont a ou b on peut définir $\mu : A^* \to \mathbf{N} \times \mathbf{N}$ par :

   $$\mu(w) = (|w|, |w|_{ab}) \ .$$

   En considérant l'ordre lexicographique $>_{lex}$ (de droite à gauche) sur $\mathbf{N} \times \mathbf{N}$, on vérifie aisément que $w \to w'$ implique $\mu(w) >_{lex} \mu(w')$.

Solution à l'exercice 2

1. [0,5 pnt] $A = (x \wedge \neg y) \vee (\neg x \wedge y)$. $B = (x \vee y) \wedge (\neg x \vee \neg y)$.
2. [1 pnt] On peut définir toutes les fonctions unaires car par exemple $1 = SSI(x, x)$, $0 = XOR(x, x)$ et $NOT(x) = XOR(x, 1)$.
3. [2 pnt] Il est facile de vérifier qu'on peut définir les 8 polynômes $p(x, y)$ suivants :

   $$0, 1, x, x+1, y, y+1, x+y, x+y+1$$

   et que ces polynômes sont stables par application des opérations *XOR* et *SSI*. Par exemple, l'addition (ou *XOR*) de deux polynômes de cette classe est encore un polynôme de la classe. Il reste donc 8 polynômes $p(x, y)$ qu'on ne peut pas définir, à savoir :

   $$xy, xy+1, xy+x+1, xy+y+1, xy+x+y, xy+x+y+1 \ .$$

4. [0,5 pnt] Voir cours.
5. [0,5 pnt]

   $$\frac{\Gamma \vdash A, \Delta \quad \Gamma \vdash \neg B, \Delta}{\Gamma \vdash A \oplus B, \Delta}$$

   — Validité. Si $v \models \Gamma$ et $v \models \Delta$ la conclusion est valide. Et si $v \not\models \Delta$ alors $v \models A$ et $v \not\models B$ et donc $v \models A \oplus B$.

   — Non réversibilité. Si $v \models \Gamma$, $v \not\models \Delta$, $v \not\models A$ et $v \models B$ la conclusion est valide mais les hypothèses ne le sont pas.

6. [1 pnt] On retrouve la CNF du point 1.

   $$\frac{\Gamma \vdash A, B, \Delta \quad \Gamma \vdash \neg A, \neg B, \Delta}{\Gamma \vdash (A \oplus B), \Delta}$$

   — Validité. Si $v \models \Gamma$ et $v \models \Delta$ la conclusion est valide. Et si $v \not\models \Delta$ alors $v \models A \vee B$ et $v \models \neg A \vee \neg B$ et donc $v \models A \oplus B$.

— Réversibilité. Si $v \models \Gamma$ et $v \models \Delta$ alors les hypothèses sont valides. Si $v \not\models \Delta$ et $v \models (A \oplus B)$ alors $v \models A \vee B$ et $v \models \neg A \vee \neg B$.

7. [1 pnt] On retrouve la DNF du point 1.

$$\frac{\Gamma, A, \neg B \vdash \Delta \quad \Gamma, \neg A, B \vdash \Delta}{\Gamma, (A \oplus B) \vdash \Delta}$$

Argument similaire à celui du point précédent.

8. [0,5 pnt] On a :

$$\frac{\vdash A, \neg A \quad \vdash \neg A, \neg\neg A}{\vdash A \oplus \neg A}$$

et on utilise la règle pour la négation et l'axiome pour dériver les deux hypothèses.

9. [0,5 pnt] Les deux hypothèses sont identiques.

$$\cfrac{\cfrac{A \vdash A}{A, \neg A \vdash}}{A \oplus A \vdash}$$

Solution à l'exercice 3

1. [1 pnt] Vrai. Si A ne contient pas de clauses positives alors chaque clause dans A doit contenir un littéral négatif et pour satisfaire ces clauses il suffit de prendre l'affectation qui associe 0 à toute variable.
2. [1 pnt] Faux. Par exemple $A = \{\{\neg x\}\}$.
3. [1 pnt] Vrai. Il suffit de prendre l'affectation qui associe 1 à toute variable.

Solution à l'exercice 4

1. [1 pnt] Avec la clause

$$(x_1 \vee \cdots \vee x_n) \tag{2}$$

on exprime la condition qu'au moins une variable doit être 1 et avec les clauses

$$(\neg x_i \vee \neg x_j) \qquad 1 \leq i < j \leq n \tag{3}$$

on exprime la condition qu'au plus une variable peut être 1. A_n est donc la conjonction de toutes ces clauses.

2. [1 pnt] Avec l'ordre $x_1 < \cdots < x_4$ le BDD réduit a 1 noeud avec étiquette x_1 et 2 noeuds avec étiquettes x_i pour $i = 2, 3, 4$ (dessin omis).
3. [0,5 pnt] Le BDD réduit a $2 \cdot n + 1$ noeuds et ce nombre ne dépend pas de l'ordre choisi.
4. [1 pnt] On peut appliquer la transformation standard pour passer d'une CNF à une 3-CNF (voir cours). Si $n > 3$, pour la clause (2), on obtient :

$$(x_1 \vee x_2 \vee y_1) \wedge (\neg y_1 \vee x_3 \vee y_2) \wedge \cdots (\neg y_{n-3} \vee x_{n-1} \vee x_n)$$

où les variables y_i sont des nouvelles variables auxiliaires. Pour les clauses (3) on obtient :

$$(\neg x_i \vee \neg x_j \vee z_{i,j}) \wedge (\neg x_i \vee \neg x_j \vee \neg z_{i,j})$$

où $z_{i,j}$ sont aussi des nouvelles variables auxiliaires.

5. [2 pnt] Oui. On remplace chaque clause $C = (\ell_1 \vee \ell_2 \vee \ell_3)$ de la 3-CNF par la H2-CNF suivante :
$$A = (\ell_1 \vee y) \wedge (\neg y \vee \ell_2 \vee \neg z) \wedge (z \vee \ell_3)$$
où y, z sont des nouvelles variables auxiliaires. Si $v \models C$ alors : (i) si $v \models \ell_1$ on prend $v(y) = 0$ et $v(z) = 1$, (ii) si $v \models \ell_2$ on prend $v(y) = 1$ et $v(z) = 1$ et (iii) si $v \models \ell_3$ on prend $v(y) = 1$ et $v(z) = 0$.
D'autre part, si $v \models A$ alors : (i) Si $v \models \ell_1$ alors $v \models C$, (ii) sinon, $v(y) = 1$ et si $v \models \ell_2$ alors $v \models C$, (iii) sinon, $v(y) = 1$, $v(z) = 0$, $v \models \ell_3$ et donc $v \models C$.
6. [1 pnt] Non. Personne (autant qu'on sache) connaît un algorithme polynomial pour H2-CNF. En effet par le point 5, on a une réduction efficace du problème de la satisfaction d'une 3-CNF au problème de la satisfaction d'une H2-CNF. Donc un algorithme efficace pour H2-CNF donnerait un algorithme efficace pour 3-CNF (et pour CNF). Notez au passage que H2-CNF est une *combinaison* (mais pas juste l'union !) de deux classes de CNF pour lesquelles on connaît des algorithmes polynomiaux en temps à savoir les 2-CNF et les formules de Horn. C'est peut être pour cette raison que votre collègue pense avoir un algorithme polynomial...