Examen - 5 juin 2014

L'examen dure 3 heures. L'énoncé est composé de 4 pages, le barème est indicatif. Toutes les réponses devront être clairement justifiées. Le seul document autorisé est une page A4 manuscripte recto-verso. Le tableau résumant les règles du système G est donné à la fin du sujet.
Inscrivez votre nom et le numéro d'anonymat sur chaque copie et numérotez-la.
Cachetez toutes les copies !

Exercice 1 *Satisfiabilité (5 points).*
Soit la formule $A \stackrel{\text{def}}{=} (\exists x, \neg P(x)) \Rightarrow ((\forall x, P(x) \lor Q(x)) \Rightarrow \forall x, Q(x))$

1. Mettre la formule A en forme normale de négation (les négations ne portent que sur les prédicats et il n'y a plus de connecteur d'implication).
2. Soit B la formule obtenue à la question précédente, dire si les affirmations suivantes sont vraies ou fausses (sans les justifier, c'est du cours !)
   — $A \models B$
   — $B \models A$
   — A est satisfiable si et seulement si B l'est.
3. Eliminer dans B les quantificateurs existentiels ($\exists$) en utilisant la skolémisation et mettre les quantificateurs universels ($\forall$) en tête de la formule.
4. Soit C la formule obtenue à la question précédente, dire si les affirmations suivantes sont vraies ou fausses (sans les justifier, c'est du cours !)
   — $B \models C$
   — $C \models B$
   — B est satisfiable si et seulement si C l'est.
5. La formule C est-elle vraie dans un modèle dont le domaine a exactement un élément ? Même question pour A, justifier vos réponses.
6. La formule A est-elle satisfiable ?
7. Mettre la formule $\neg A$ en forme clausale.
8. Indiquer le domaine et la base de Herbrand pour la forme clausale ainsi obtenue.
9. La formule $\neg A$ est-elle satisfiable ? La formule A est-elle valide ?

Exercice 2 *Logique propositionnelle (3 points).*
Soit les séquents

S_1 **:** $(A \Rightarrow (C \Rightarrow B)) \Rightarrow (A \land C) \vdash A$

S_2 **:** $(A \Rightarrow (C \Rightarrow B)) \Rightarrow (A \land C), A \Rightarrow C \vdash$

1. Donner les formules logiques associées à ces séquents.
2. Construire les tables de vérité des deux formules précédentes.

3. Peut-on construire des preuves complètes dans le système G des séquents donnés ? si oui construire la preuve, sinon expliquer pourquoi.

Exercice 3 *Modélisation, (7 points).*

On cherche à modéliser un championnat de football. Pour cela on se place dans un univers dans lequel les objets représentent des équipes ou des résultats de match. Chaque équipe affronte une autre équipe deux fois, une fois dans son propre stade et une fois dans le stade de son adversaire. Le résultat d'un match est le résultat de l'équipe qui accueille le match, cela peut être gagné, perdu ou match nul.

On introduit les symboles suivants :

— trois constantes `G`, `P`, `N` qui représentent les résultats possibles des matchs : `G` si le match est gagné, `P` s'il est perdu et `N` s'il est nul ; ces constantes sont distinctes

— `equipe`, prédicat unaire : `equipe`(e) représente le fait que e est une équipe ;

— `résultat`, un prédicat ternaire `résultat`(e_1, e_2, r) représente le fait que le match de l'équipe e_1 avec l'équipe e_2 sur le stade de e_1 a été joué avec le résultat r.

— une relation binaire d'égalité.

On fera attention au fait qu'une équipe e gagne si le résultat du match sur son terrain est `G` ou bien si le résultat du match sur le terrain de l'adversaire est `P` et que ne pas gagner ne signifie pas que l'on a perdu (un match nul n'est ni perdu, ni gagné).

1. Exprimer à l'aide de formules logiques les propriétés suivantes :
   (a) il y a au plus un résultat pour chaque match ;
   (b) une équipe ne peut pas jouer contre elle-même ;
   (c) aucune équipe ne perd sur son propre terrain ;
   (d) toute équipe se fait battre au moins une fois ;
2. On se place dans des modèles avec quatre équipes A, F, I et E. Les tableaux suivants représentent les valeurs du prédicat `resultat` : on a la valeur r sur la ligne de l'équipe e_1 et la colonne de l'équipe e_2 si et seulement si le prédicat `resultat`(e_1, e_2, r) est vrai dans le modèle.

	A	F	I	E
A		N	G	G
F	N		N	G
I	N	N		N
E	G	N	G	

modèle 1

	A	F	I	E
A		G	G	G
F	G		N	N
I		N		P
E	G	N	G	

modèle 2

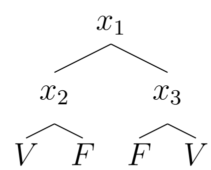
   Pour chaque formule 1c, 1d de la question 1, dire si elles sont vraies ou fausses dans chacun des deux modèles 1 et 2 ci-dessus.

3. Exprimer à l'aide d'une formule logique qui contient une variable libre e les faits suivants :
   — formule $A(e)$: l'équipe e a gagné tous les matchs qu'elle a joués
   — formule $B(e)$: l'équipe e n'a perdu aucun match
   — formule $C(e)$: l'équipe e a gagné au moins un match
4. On reprend les formules $A(e)$, $B(e)$ et $C(e)$ de la question 3. Dire si les formules suivantes sont valides ou non, si elles sont valides on le justifiera par des preuves ou des équivalences logiques, sinon on proposera un modèle dans lequel la formule est fausse.

(a) $A(e) \Rightarrow B(e)$
(b) $B(e) \Rightarrow A(e)$
(c) $\neg A(e) \Rightarrow \neg B(e)$
(d) $B(e) \Rightarrow C(e)$

Exercice 4 *Arbres de décisions binaires (3 points).*
On représente des formules du calcul propositionnel avec des variables $x_1, \ldots, x_n$ par des arbres de décision binaire.

Un *arbre de décision binaire* est un arbre binaire dont les nœuds sont étiquetés par les variables propositionnelles et les feuilles sont formées des constantes booléennes V ou F. On se restreindra de plus aux arbres tels que l'indice des variables propositionnelles des fils est strictement plus grand que celui du père. Par exemple, on notera `bdt` l'arbre suivant



On représente l'arbre en utilisant un constructeur `Bool` pour les feuilles, associé à une valeur booléenne V ou F et un constructeur à trois arguments $\mathbf{IF}(i, P, Q)$ avec i un entier pour repésenter la variable propositionnelle x_i et P et Q des arbres de décision binaire. Pour l'exemple `bdt` cela donne $\mathbf{IF}(1, \mathbf{IF}(2, \texttt{Bool}(V), \texttt{Bool}(F)), \mathbf{IF}(3, \texttt{Bool}(F), \texttt{Bool}(V)))$.
Un arbre de décision binaire t représente une formule propositionnelle $\texttt{form}(t)$ définie par les équations suivantes :

$$\begin{aligned}
&\texttt{form}(\texttt{Bool}(V)) = \top \\
&\texttt{form}(\texttt{Bool}(F)) = \bot \\
&\texttt{form}(\mathbf{IF}(i, P, Q)) = (x_i \wedge \texttt{form}(P)) \vee (\neg x_i \wedge \texttt{form}(Q))
\end{aligned}$$

La valeur d'un arbre de décision binaire dans une interprétation I des variables propositionnelles est donnée par les équations suivantes

$$\begin{aligned}
&\texttt{vald}(I, \texttt{Bool}(b)) = b \\
&\texttt{vald}(I, \mathbf{IF}(i, P, Q)) = \texttt{vald}(I, P) \quad \text{si } I(x_i) = V \\
&\texttt{vald}(I, \mathbf{IF}(i, P, Q)) = \texttt{vald}(I, Q) \quad \text{si } I(x_i) = F
\end{aligned}$$

1. Donner la formule propositionnelle associée à l'arbre `bdt` de l'exemple.
2. Construire la table de vérité de la formule associée à l'arbre `bdt` de l'exemple.
3. Donner les équations récursives d'une fonction `check` qui prend en argument un entier k et un arbre de décision binaire t et vérifie qu'il satisfait la condition de croissance des indices des variables propositionnelles le long des branches et que toutes les variables dans l'arbre ont un indice strictement supérieur à k.
4. Donner les équations récursives d'une fonction `tauto` qui prend en argument un arbre de décision binaire t et vérifie que c'est une tautologie.

5. Compléter les équations récursives ci-dessous pour définir une fonction `impl` qui prend en argument deux arbres de décision binaire t_1 (qui représente la formule P_1) et t_2 (qui représente la formule P_2) qui vérifient la condition de croissance et construit l'arbre de décision binaire correspondant à la formule $P_1 \Rightarrow P_2$.

$$
\begin{array}{lll}
\texttt{impl}(\texttt{Bool}(V), P) & = & \\
\texttt{impl}(P, \texttt{Bool}(V)) & = & \\
\texttt{impl}(\texttt{Bool}(F), P) & = & \\
\texttt{impl}(\mathbf{IF}(i, P, Q), A) & = & \qquad\qquad\qquad \text{si } A = \texttt{Bool}(F) \\
\texttt{impl}(\mathbf{IF}(i, P, Q), \mathbf{IF}(j, P', Q')) & = & \qquad\qquad\qquad \text{si } i = j \\
\texttt{impl}(\mathbf{IF}(i, P, Q), \mathbf{IF}(j, P', Q')) & = & \qquad\qquad\qquad \text{si } i < j \\
\texttt{impl}(\mathbf{IF}(i, P, Q), \mathbf{IF}(j, P', Q')) & = & \qquad\qquad\qquad \text{si } j < i
\end{array}
$$

Exercice 5 *Résolution (2 points).*
On se place dans un langage avec deux symboles de constantes `G` et `P` et un prédicat ternaire R. Soient les quatre formules de la logique du premier ordre suivantes :
— F_1 : $\exists x, \forall y, \neg R(x, y, \texttt{P}) \wedge \neg R(y, x, \texttt{G})$
— F_2 : $\forall x\, y\, z, R(x, y, z) \Rightarrow (R(x, y, \texttt{P}) \vee R(x, y, \texttt{G}))$
— F_3 : $\forall x, \exists y, \exists z, R(x, y, z)$
— F_4 : $\exists x, \exists y, R(x, y, \texttt{G})$

1. Mettre en forme clausale les formules F_1, F_2, F_3 et $\neg F_4$. On prendra soin d'expliciter les symboles de skolem introduits.
2. Montrez **en utilisant la méthode de résolution** que la formule F_4 est conséquence logique des formules F_1, F_2 et F_3.

Rappel des règles logiques du système G

hypothèse	$(\textsc{Hyp})\dfrac{}{A, \Gamma \vdash \Delta, A}$	
	gauche	droite
$\bot$	$(\textsc{Jok})\dfrac{}{\bot, \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \bot}$
$\top$	$\dfrac{\Gamma \vdash \Delta}{\top, \Gamma \vdash \Delta}$	$(\textsc{Triv})\dfrac{}{\Gamma \vdash \Delta, \top}$
$\neg$	$\dfrac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}$	$\dfrac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}$
$\wedge$	$\dfrac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}$
$\vee$	$\dfrac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}$
$\Rightarrow$	$\dfrac{\Gamma \vdash \Delta, A \quad B, \Gamma \vdash \Delta}{A \Rightarrow B, \Gamma \vdash \Delta}$	$\dfrac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \Rightarrow B}$