

Exercice 1 : Preuve dans le système LP (4 points)

Le système formel LP est composé des axiomes suivants et de la règle de modus ponens, supposée connue :

- Axiome 1 : $P \Rightarrow (Q \Rightarrow P)$
 Axiome 2 : $(P \Rightarrow Q) \Rightarrow ((P \Rightarrow (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R))$
 Axiome 3 : $P \Rightarrow (Q \Rightarrow P \wedge Q)$
 Axiome 4 : $P \wedge Q \Rightarrow P$
 Axiome 5 : $P \wedge Q \Rightarrow Q$
 Axiome 6 : $P \Rightarrow P \vee Q$
 Axiome 7 : $Q \Rightarrow P \vee Q$
 Axiome 8 : $(P \Rightarrow R) \Rightarrow ((Q \Rightarrow R) \Rightarrow (P \vee Q \Rightarrow R))$
 Axiome 9 : $\neg\neg P \Rightarrow P$
 Axiome 10 : $(P \Rightarrow Q) \Rightarrow ((P \Rightarrow \neg Q) \Rightarrow \neg P)$

Présenter sous forme de tableau une preuve formelle de $\neg A \Rightarrow B, \neg B \vdash A$ dans le système LP.

(4 points : 0,5 point par ligne, sauf la dernière) Dans le système formel LP, une démonstration complète de ce théorème est :

1	Hyp. 1	$\neg A \Rightarrow B$
2	Ax. 10 ($\neg A/P, B/Q$)	$(\neg A \Rightarrow B) \Rightarrow ((\neg A \Rightarrow \neg B) \Rightarrow \neg\neg A)$
3	m.p. sur 1, 2	$(\neg A \Rightarrow \neg B) \Rightarrow \neg\neg A$
4	Hyp. 2	$\neg B$
5	Ax. 1 ($\neg B/P, \neg A/Q$)	$\neg B \Rightarrow (\neg A \Rightarrow \neg B)$
6	m.p. sur 4, 5	$\neg A \Rightarrow \neg B$
7	m.p. sur 6, 3	$\neg\neg A$
8	Ax. 9 (A/P)	$\neg\neg A \Rightarrow A$
9	m.p. sur 7, 8	A

Exercice 2 : Résolution propositionnelle (5 points)

1. Définir toutes les variables propositionnelles utiles pour formaliser les énoncés suivants :

- (a) Quand un étudiant résout un exercice sans soupirer, c'est qu'il le comprend.
- (b) Ces exercices ne sont pas sous une forme habituelle.
- (c) Les exercices faciles ne donnent pas mal à la tête.
- (d) Les étudiants ne comprennent pas les exercices qui ne sont pas sous une forme habituelle.
- (e) Un étudiant ne soupire jamais devant un exercice, sauf s'il lui donne mal à la tête.

(1 point) On introduit les propositions suivantes :

- S : « l'étudiant résout un exercice en soupirant »,
- C : « l'étudiant comprend cet exercice »,
- H : « cet exercice se présente sous une forme habituelle »,
- F : « cet exercice est facile »,
- M : « cet exercice donne mal à la tête ».

2. En utilisant ces variables, traduire séparément chacun de ces énoncés en une formule propositionnelle.

(2,5 points : 0,5 point par énoncé) Les énoncés se formalisent alors par

- (a) $\neg S \Rightarrow C$
- (b) $\neg H$
- (c) $F \Rightarrow \neg M$
- (d) $\neg H \Rightarrow \neg C$
- (e) $S \iff M$

3. Appliquez la méthode de résolution propositionnelle, de manière détaillée, pour décider si la conjonction de ces énoncés est contradictoire ou non. **Attention : l'application de toute autre méthode de raisonnement est exclue. Elle ne rapporterait aucun point.**

(1,5 point)

La mise en CNF de ces formules donne les 6 clauses suivantes : $C_1 = S \vee C$, $C_2 = \neg H$, $C_3 = \neg F \vee \neg M$, $C_4 = H \vee \neg C$, $C_5 = \neg S \vee M$ et $C_6 = \neg M \vee S$.

En appliquant répétitivement la règle de résolution propositionnelle aux clauses de C_1 à C_6 , on produit les nouvelles clauses suivantes :

$$C_7 = \text{Res}(C_2, C_4) = \neg C,$$

$$C_8 = \text{Res}(C_7, C_1) = S,$$

$$C_9 = \text{Res}(C_5, C_8) = M$$

et

$$C_{10} = \text{Res}(C_3, C_9) = \neg F.$$

Après élimination des clauses plus faibles, il reste l'ensemble $\{\neg H, M, S, \neg C, \neg F\}$, qui est saturé et qui ne contient pas la clause vide (\perp). **Donc la conjonction de ces énoncés n'est pas contradictoire.**

Exercice 3 : Formalisation en logique des prédicats (3 points)

Dans cet exercice, on désigne par 0 et 1 les deux premiers entiers naturels, on désigne par + et \times l'addition et la multiplication de deux entiers naturels, et on désigne par = et \leq l'égalité et la relation d'ordre entre les entiers naturels.

En utilisant ces symboles et les notations de la logique des prédicats, formaliser les propriétés suivantes :

1. Il n'existe pas de plus grand entier.
2. Certains entiers sont des nombres pairs.
3. Certains entiers n'ont pas de racine carrée entière.

1. Il n'existe pas de plus grand entier : $\neg(\exists x. (\forall y. y \leq x))$.
2. Certains entiers sont des nombres pairs : $\exists x. (\exists k. x = k + k)$.
3. Certains entiers n'ont pas de racine carrée entière : $\exists x. (\forall y. \neg(x = y \times y))$.

Exercice 4 : Associativité de l'addition (3 points)

Le type inductif \mathbb{N} des entiers naturels est défini par les règles

$$\frac{}{0 : \mathbb{N}} (\mathbb{N}_0) \quad \text{et} \quad \frac{n : \mathbb{N}}{S(n) : \mathbb{N}} (\mathbb{N}_S)$$

où $S(n)$ désigne le successeur de n , c'est-à-dire $n + 1$. L'addition de deux entiers naturels est calculée par la fonction *plus* définie par les égalités suivantes :

$$\text{plus}(x, 0) = x \tag{1}$$

$$\text{plus}(x, S(i)) = S(\text{plus}(x, i)) \tag{2}$$

Prouver formellement, par induction sur c , la propriété

$$\text{plus}(a, \text{plus}(b, c)) = \text{plus}(\text{plus}(a, b), c)$$

d'associativité de l'addition, pour tous les entiers naturels a , b et c , **en détaillant chaque étape, et en indiquant le numéro de l'égalité utilisée dans chaque étape.**

Preuve inductive de la propriété $plus(a, plus(b, c)) = plus(plus(a, b), c)$ d'associativité de l'addition, par induction sur c .

— (1 point) Premier cas : $c = 0$

$$= plus(a, plus(b, 0))$$

= par (1)

$$plus(a, b)$$

et

$$plus(plus(a, b), 0)$$

$$= plus(a, b)$$

par (1)

— (2 points) Deuxième cas : $c = S(d)$

$$= plus(a, plus(b, S(d)))$$

$$= plus(a, S(plus(b, d)))$$

par (2)

$$= S(plus(a, plus(b, d)))$$

$$= S(plus(a, plus(b, d)))$$

par (2)

$$= S(plus(a, plus(b, d)))$$

Hypothèse d'induction

$$= S(plus(plus(a, b), d))$$

et

$$plus(plus(a, b), S(d))$$

$$= S(plus(plus(a, b), d))$$

par (2)

$$S(plus(plus(a, b), d))$$

Exercice 5 : Types et calculs inductifs (5 points)

1. Les constructeurs *Nil* et *Cons* du type inductif $liste(\alpha)$ des listes dont tous les éléments sont de type α peuvent être considérés comme des fonctions.

Donner leur type en tant que fonctions.

Le type inductif $liste(\alpha)$ des listes dont tous les éléments sont de type α est défini par les règles

$$\frac{}{Nil : liste(\alpha)} \quad \text{et} \quad \frac{a : \alpha, l : liste(\alpha)}{Cons(a, l) : liste(\alpha)},$$

donc on a les hypothèses de typage suivantes :

(1 point, question de cours)

$$\begin{array}{l} (H_{Nil}) \quad Nil : liste(\alpha) \\ (H_{Cons}) \quad Cons : \alpha \times liste(\alpha) \rightarrow liste(\alpha) \end{array}$$

2. Expliquer clairement et simplement pourquoi il n'est pas possible de définir une fonction de type $liste(\alpha) \rightarrow \alpha$, qui calculerait le premier élément de toute liste.

(1 point) La liste vide n'ayant pas de premier élément, on ne peut pas définir la valeur de cette fonction pour le constructeur *Nil*.

3. Pour tout type α , le type inductif $option(\alpha)$ est défini par les règles

$$\frac{}{None : option(\alpha)} (o_0) \quad \text{et} \quad \frac{x : \alpha}{Some(x) : option(\alpha)} (o_S).$$

Avec la définition du type \mathbb{N} donnée plus haut dans ce sujet, démontrer formellement que l'expression $Some(S(0))$ est de type $option(\mathbb{N})$.

Dans le sujet d'examen, la deuxième règle donnée était

$$\frac{x : \alpha}{Some(c) : option(\alpha)}$$

au lieu de (o_S) . L'évaluation a tenu compte de cette erreur dans l'énoncé.

(1 point) Une preuve formelle que l'expression $Some(S(0))$ est de type $option(\mathbb{N})$ est

$$\frac{\frac{\frac{}{0 : \mathbb{N}}}{S(0) : \mathbb{N}}}{Some(S(0)) : option(\mathbb{N})} \quad \begin{array}{l} (N_0) \\ (N_S) \\ (o_S) \end{array}$$

4. Définir une fonction

$$fst : liste(\alpha) \rightarrow option(\alpha) \quad (3)$$

qui retourne *None* pour la liste vide, et sinon calcule le premier élément de toute liste non vide, et le convertit en un habitant du type $option(\alpha)$ à l'aide du constructeur *Some*.

(1 point)

$$fst(Nil) = None \quad (4)$$

$$fst(Cons(a, l)) = Some(a) \quad (5)$$

5. On rappelle la règle suivante d'inférence et de vérification de type :

$$\frac{f : \alpha_1 \times \dots \times \alpha_n \rightarrow \beta, e_1 : \alpha_1, \dots, e_n : \alpha_n}{f(e_1, \dots, e_n) : \beta} \quad (application)$$

Avec l'hypothèse

$$(H_=) \quad = : \alpha \times \alpha \rightarrow \mathbb{B}$$

sur le type de l'égalité, avec toutes les règles dans cet énoncé et avec d'éventuelles hypothèses supplémentaires, donner une preuve formelle par arbre que chaque égalité de votre réponse à la question 4 a le type \mathbb{B} . **Indiquer les hypothèses supplémentaires nécessaires pour que ces preuves soient complètes.**

(1 point) Preuve de typage de l'égalité (4) :

$$\frac{(H_=), \frac{(3), (H_{Nil})}{fst(Nil) : option(\alpha)}, (application) \quad \frac{}{None : option(\alpha)} \quad (o_0)}{fst(Nil) = None : \mathbb{B}} \quad (application)$$

Les hypothèses supplémentaires nécessaires pour cette preuve sont (3) et (H_{Nil}) .

Preuve de typage de l'égalité (5) :

$$\frac{(H_=), \frac{(3), \frac{H_{Cons}, a : \alpha, l : liste(\alpha)}{Cons(a, l) : liste(\alpha)} \quad (app.)}{fst(Cons(a, l)) : option(\alpha)} \quad (app.) \quad \frac{a : \alpha}{Some(a) : option(\alpha)} \quad (o_S)}{fst(Cons(a, l)) = Some(a) : \mathbb{B}} \quad (app.)$$

Les hypothèses supplémentaires nécessaires pour cette preuve sont (3), $H_{Cons}, a : \alpha$ et $l : liste(\alpha)$.