DEA MDFI

Examen de logique

Exercice 1 On rappelle que la négation est définie par $\neg A = A \to \perp$.

i) Donner une preuve en déduction naturelle de $\neg\neg(A \vee \neg A)$.

ii) Donner une preuve en déduction naturelle classique de $(A \to B) \vee (B \to A)$.

iii) Montrer que cette dernière formule n'est pas prouvable intuitionnistiquement.

Exercice 2 Soient $(a_n)_{n\in\mathbb{N}}$ et $(b_n)_{n\in\mathbb{N}}$ deux familles de points deux à deux distincts et N l'espace cohérent dont la trame est $\{a_n, n \in \mathbb{N}\} \cup \{b_n, n \in \mathbb{N}\}$ et la relation de cohérence est donnée par :

$$\begin{array}{ll} a_n \frown a_p & \text{pour tous } n \text{ et } p \\ a_n \frown b_p & \text{ssi } n < p \\ b_n \frown b_p & \text{ssi } n = p \end{array}$$

i) Donner toutes les relations d'incohérence sur la trame de N.

ii) Montrer que pour chaque entier n il existe une unique clique maximale $\widetilde{n}$ qui contient exactement $n+1$ éléments.

iii) Trouver une fonction stable f des cliques de N dans les cliques de N telle que $f(\widetilde{n}) = \widetilde{n+1}$ pour tout n. Donner la trace de f.

iv) Dire pour chacun des ensembles suivants s'ils sont des traces de fonctions stables de N dans N et si c'est le cas, calculer l'image de la fonction sur les cliques $\widetilde{n}$;

$$\begin{array}{cc} \{(\emptyset, b_1)\} & \{(\emptyset, a_0), (\emptyset, b_1)\} \\ \{(\emptyset, b_0), (\emptyset, b_1)\} & \{(\{a_0\}, a_0), (\{a_1\}, a_0)\} \\ \{(\{a_0\}, a_0), (\{a_1\}, b_1)\} & \{(\{a_0\}, a_0), (\{a_0\}, b_1)\} \\ \{(\{a_0\}, b_0), (\{a_1\}, b_1)\} & \{(\{b_0\}, a_0), (\{b_1\}, b_1)\} \end{array}$$

Exercice 3 On rappelle que $\mathbb{N}$ est le type des entiers de Church $\forall\alpha((\alpha \to \alpha) \to \alpha \to \alpha)$, que pour tout entier k, on note $\underline{k}$ l'entier de Church représentant k et que la définition de la fonction d'Ackermann est :

$$\begin{aligned} A(0,p) &= p+1 \\ A(n+1,0) &= A(n,1) \\ A(n+1,p+1) &= A(n, A(n+1,p)) \end{aligned}$$

Le but de cet exercice est de représenter A par un lambda-terme typable dans le système F, c'est à dire de trouver un lambda-terme $\underline{A}$ typable de type $\mathbb{N} \to \mathbb{N} \to \mathbb{N}$ tel que pour tous entiers n et p on ait $(\underline{A})\underline{n}\,\underline{p} =_\beta \underline{A(n,p)}$.

i) Pour chaque n on note A_n la fonction définit par $A_n(p) = A(n,p)$. Exprimer $A_{n+1}(p)$ en fonction de A_n et de p. En déduire une fonctionnelle B telle que pour tout n on ait $B(A_n) = A_{n+1}$ puis un terme $\underline{B}$ représentant B c'est à dire typable de type $(\mathbb{N} \to \mathbb{N}) \to \mathbb{N} \to \mathbb{N}$ et tel que pour tout n si $\underline{A_n}$ est un terme représentant A_n alors $(B)\underline{A_n}$ représente A_{n+1}.

ii) Écrire $A(n,p)$ directement en fonction de B, n, p et A_0. En déduire le terme $\underline{A}$ cherché (dont on n'oubliera pas de montrer qu'il est typable du bon type).

Exercice 4 On suppose définie une fonction de codage des preuves en déduction naturelle du second ordre (système F), c'est à dire qu'à toute déduction π on sait associer un entier $\ulcorner\pi\urcorner$ et que l'on a un prédicat primitif récursif $\mathrm{Pr}(x)$ équivalent à «x est le code d'une preuve dans le système F».

i) En utilisant le fait (non démontré en cours mais bien connu tout de même) que la fonction d'Ackermann n'est pas primitive récursive, montrer que la fonction f de normalisation du système F n'est pas primitive récursive. La fonction f est définie par :

$$f(n) = \begin{cases} \ulcorner\pi_0\urcorner & \text{si } n \text{ est le code d'une preuve } \pi \text{ dont la forme normale est } \pi_0\,; \\ 0 & \text{sinon.} \end{cases}$$