I Exercice CCP

Rappelons les règles de déduction naturelle suivantes, où A et B sont des formules logiques et Γ un ensemble de formules logiques quelconques :

$$\frac{}{\Gamma, A \vdash A}\text{AX} \qquad \frac{\Gamma \vdash \bot}{\Gamma \vdash A}\bot_e \qquad \frac{\Gamma, A \vdash B}{\Gamma \vdash A \to B}\to_i \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash A \to B}{\Gamma \vdash B}\to_e \qquad \frac{\Gamma, A \vdash \bot}{\Gamma \vdash \neg A}\neg_i \qquad \frac{\Gamma \vdash A \qquad \Gamma \vdash \neg A}{\Gamma \vdash \bot}\neg_e$$

1. Montrer que le séquent $\vdash \neg A \to (A \to \bot)$ est dérivable, en explicitant un arbre de preuve.
2. Montrer que le séquent $\vdash (A \to \bot) \to \neg A$ est dérivable, en explicitant un arbre de preuve.
3. Donner une règle correspondant à l'introduction du symbole $\wedge$ ainsi que deux règles correspondant à l'élimination du symbole $\wedge$. Montrer que le séquent $\vdash (\neg A \to (A \to \bot)) \wedge ((A \to \bot) \to \neg A)$ est dérivable.
4. On considère la loi de Peirce $P = ((A \to B) \to A) \to A$. Montrer que $\models P$, c'est-à-dire que P est une tautologie.
5. Pour montrer que le séquent $\vdash P$ est dérivable, il est nécessaire d'utiliser la règle d'absurdité classique $\bot_c$ (ou une règle équivalente), ce que l'on fait ci-dessous (il n'y aura pas besoin de réutiliser cette règle). Terminer la dérivation du séquent $\vdash P$, dans laquelle on pose $\Gamma = \{(A \to B) \to A, \neg A\}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{?}{\Gamma \vdash A}\,? \qquad \dfrac{}{\Gamma \vdash \neg A}\,\text{AX}}{\Gamma = (A \to B) \to A, \neg A \vdash \bot}\,\neg_i}{(A \to B) \to A) \vdash A}\,\bot_c}{\vdash ((A \to B) \to A) \to A}\,\to_i$$

II Lois de de Morgan

1. Prouver le séquent $\neg p \vee \neg q \vdash \neg(p \wedge q)$.
2. Prouver le séquent $\neg(p \vee q) \vdash \neg p \wedge \neg q$.
3. Prouver le séquent $\neg p \wedge \neg q \vdash \neg(p \vee q)$.
4. En utilisant le tiers exclu de la logique classique $\dfrac{}{\Gamma \vdash p \vee \neg p}\text{te}$, prouver le séquent $\neg(p \wedge q) \vdash \neg p \vee \neg q$.

III Complétude de la logique classique

On souhaite montrer dans cet exercice que la logique classique est complète. On suppose $V = \{x_1, , x_n\}$ l'ensemble des variables propositionnelles. Pour A une formule et μ une valuation, on note :

$$|A|_\mu = \begin{cases} A & \text{si } \mu(A) = 1 \\ \neg A & \text{sinon} \end{cases}$$

1. Soit A une formule et μ une valuation. On note $\Gamma = \{|x_1|_\mu, |x_2|_\mu, , |x_n|_\mu\}$. Montrer par induction structurelle sur A que $\Gamma \vdash |A|_\mu$.
2. Soit x une variable et Γ un contexte quelconque. Montrer que si $\Gamma, x \vdash A$ et $\Gamma, \neg x \vdash A$, alors $\Gamma \vdash A$.
3. En déduire que si A est une tautologie, alors A est un théorème.
4. En déduire la complétude de la logique classique : si $\Gamma \models A$ alors $\Gamma \vdash A$ est prouvable.

IV Quantificateurs

Montrer les séquents suivants :

1. $\vdash \forall x\, A \to \exists x\, A$.
2. $\exists x\, \neg A \vdash \neg(\forall x\, A)$.

V Typage OCaml

On souhaite formaliser le typage OCaml. Pour cela, on notera $\Gamma \vdash e : \tau$ si l'expression OCaml e est typée par le type τ et on utilisera les règles suivantes :

$$\frac{}{\Gamma \vdash \texttt{false} : \texttt{bool}}\ (1) \qquad \frac{}{\Gamma \vdash \texttt{true} : \texttt{bool}}\ (2) \qquad \frac{n \in \mathbb{N}}{\Gamma \vdash n : \texttt{int}}\ (3)$$

$$\frac{}{\Gamma, x : \tau \vdash x : \tau}\ (4) \qquad \frac{\Gamma, x : \sigma \vdash e : \tau}{\Gamma \vdash \texttt{fun}\ x \to e : \sigma\ \to\ \tau}\ (5) \qquad \frac{\Gamma \vdash f : \sigma\ \to\ \tau \qquad \Gamma \vdash e : \sigma}{\Gamma \vdash f\ e : \tau}\ (6)$$

1. Soit $\Gamma = \{$`f : a -> (b -> a), g : b -> a`$\}$. Montrer $\Gamma \vdash$ `fun x -> f (g x) x` $: \tau$ pour un certain type τ à déterminer.
2. Quelles analogies peut-on faire entre le typage OCaml et la déduction naturelle ?
3. Montrer que `(fun g -> g 1 2) (fun x -> 3)` n'est pas typable, c'est-à-dire qu'il n'existe pas de type τ tel que $\vdash$ `(fun g -> g 1 2) (fun x -> 3)` $: \tau$ soit prouvable.

On ajoute maintenant les tuples :

$$\frac{\Gamma \vdash e_1 : \tau_1 \qquad \Gamma \vdash e_2 : \tau_2}{\Gamma \vdash (e_1, e_2) : \tau_1 * \tau_2}$$

On veut aussi ajouter des fonctions polymorphes.

4. En utilisant des quantificateurs, proposer des types pour `fst` et `snd`, et une règle d'élimination.
5. Montrer alors que `fst (42, true)` est bien typé.