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 $\Gamma = \{(A \to B) \to A, \neg A\}$:

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

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 $\frac{}{\Gamma \vdash p \vee \neg p}\text{te}$, 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 note $V = \{x_1, ..., x_n\}$ l'ensemble des variables propositionnelles. Pour A une formule et v une valuation, on note :

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

1. Soit A une formule et v une valuation. On note $\Gamma = \{|x_1|_v, |x_2|_v, ..., |x_n|_v\}$. Montrer par induction structurelle sur A que $\Gamma \vdash |A|_v$.
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.

Déduction de messages

Nous souhaitons nous intéresser au problème suivant appelé problème de déduction :

entrée un ensemble fini de termes clos T et un terme clos u

sortie est-ce que u est déductible depuis T, noté $T \vdash u$?

Terme et sous-terme Nous nous intéressons aux termes construits inductivement à partir du symbole binaire $f(\cdot,\cdot)$, d'un ensemble infini dénombrable de constantes $\mathcal{C}$, et d'un ensemble infini dénombrable de variables $\mathcal{V}$.
Un terme est donc généré par la grammaire : $t, t_1, t_2 := v \in \mathcal{C} \mid x \in \mathcal{V} \mid f(t_1, t_2)$.
Si un terme ne contient pas de variable, alors ce terme est dit *clos*.
Étant donné un terme t nous notons $st(t)$ l'ensemble des *sous-termes* de t, i.e., le plus petit ensemble S tel que $t \in S$, et si $f(t_1, t_2) \in S$ alors $t_1, \ldots, t_n \in S$.

Règle d'inférence une règle d'inférence est une règle de déduction de la forme :

$$\frac{T \vdash t_1 \quad \ldots \quad T \vdash t_n}{T \vdash t}\ \textsc{Regle}$$

où T est un ensemble fini de termes et $t_1, \ldots, t_n, t$ sont des termes.
Un *système d'inférence* $\mathcal{I}$ est un ensemble fini de règles d'inférence.

Preuve Une *preuve* (ou *arbre de preuve*) Π de $T \vdash u$ dans $\mathcal{I}$ est un arbre tel que :

— chaque feuille est étiquetée avec un terme $v \in T$;

— pour chaque noeud ayant pour étiquette v_0 et enfants $v_1, \ldots, v_n$ il existe une règle d'inférence dans $\mathcal{I}$ ayant pour conclusion v_0 et hypothèses $v_1, \ldots, v_n$ (à instanciation près des variables) ;

— la racine de l'arbre est étiquetée par u.

La taille d'une preuve Π, notée *size*(Π), est son nombre de noeuds. $Termes(\Pi)$ dénote l'ensemble des étiquettes, i.e., termes, apparaissant dans Π.

Lorsque $T \vdash u$ nous disons que u est déductible à partir de l'ensemble de termes T.

$$\frac{\text{si } u \in T}{T \vdash u}\ \textsc{Ax}$$

$$\frac{T \vdash x \quad T \vdash y}{T \vdash f(x,y)}\ \textsc{App-f} \qquad \frac{T \vdash f(x,y) \quad T \vdash y}{T \vdash x}\ \textsc{Red-f}$$

FIGURE 1 – Système d'inférence $\mathcal{I}_0$

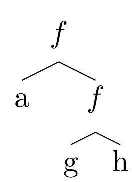

FIGURE 2 – Représentation sous forme d'arbre du terme $f(a, f(g, h))$

Question 1. Soit $T = \{f(f(a, k_1), k_2), k_2, f(k_1, k_2)\}$. Donner l'arbre de preuve de $T \vdash a$ dans $\mathcal{I}_0$.

Question 2. Soit T un ensemble de termes clos. Montrer que pour tout $T \vdash u$ dans $\mathcal{I}_0$, un arbre de preuve de taille minimale Π de $T \vdash u$ contient seulement des termes issus de $st(T \cup \{u\})$, i.e., $Termes(\Pi) \subseteq st(T \cup \{u\})$.

Montrer de plus que si Π est réduit à une feuille ou termine par une règle Ax ou Red-f alors il contient uniquement des termes issus de $st(T)$, i.e. $Termes(\Pi) \subseteq st(T)$.

Question 3. En déduire que le problème de déduction dans $\mathcal{I}_0$ est décidable en temps polynomial.

Nous considérerons que la taille du problème est : $size(T, u) = |st(u)| + \Sigma_{t \in T} |st(t)|$.

On définit le problème HORN-SAT :

entrée une formule Φ étant une conjonction finie de clauses de Horn

sortie est-ce que Φ satisfiable ?

Une clause de Horn est une formule du calcul propositionnel qui contient au plus un littéral positif. Une clause de Horn peut donc avoir trois formes :

— un littéral positif et aucun négatif : $C = (true \Rightarrow x)$

— un littéral positif et au moins un littéral négatif : $C = (x_1 \wedge \ldots \wedge x_n \Rightarrow x)$

— aucun littéral positif : $C = (x_1 \wedge \ldots \wedge x_n \Rightarrow false)$.

On admettra que HORN-SAT est P-complet, c'est-à-dire (intuitivement) que tout problème de décision dans P admet une réduction linéaire à HORN-SAT.

Question 4. Montrer que le problème de déduction dans $\mathcal{I}_0$ est P-complet.

Nous souhaiterions maintenant nous intéresser au même problème mais en ajoutant le ou-exclusif. Un terme est donc maintenant généré par la grammaire :

$$t, t_1, t_2 := v \in \mathcal{C} \mid x \in \mathcal{V} \mid f(t_1, t_2) \mid t_1 \oplus t_2.$$

Nous ne prouverons pas ici que le problème de déduction est encore décidable en temps polynomial. Nous nous intéresserons à prouver une étape de la preuve : étant donné un ensemble de termes T et un terme t, est-ce que $T \vdash t$ en utilisant uniquement les règles GX et Ax' ?

$$\frac{T \vdash u_1 \quad \ldots \quad T \vdash u_n}{T \vdash u_1 \oplus \ldots \oplus u_n} \text{ GX} \qquad \frac{\text{si } u =_{AC} v \quad \text{et } v \in T}{T \vdash u} \text{ Ax'}$$

On note $=_{AC}$ la plus petite relation telle que :

(refl.) $x = x$ (sym.) $(x = y) \Rightarrow (y = x)$ (trans.) $(x = y) \wedge (y = z) \Rightarrow (x = z)$

(comm.) $x \oplus y = x \oplus y$ (assoc.) $x \oplus (y \oplus z) = (x \oplus y) \oplus z$

(congr.) $(x_1 = y_1) \wedge (x_2 = y_2) \Rightarrow f(x_1, x_2) = f(y_1, y_2)$

Question 5. Soit u et v deux termes clos. Donner un algorithme en temps polynomial qui décide si $u =_{AC} v$.

Question 6. Soit T un ensemble de termes clos. Soit t un terme clos.
Montrer que $T \vdash t$ dans $\{\mathrm{GX}, \mathrm{Ax'}\}$ est décidable en temps polynomial.