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.

   Solution :
   $$\dfrac{\dfrac{\dfrac{\dfrac{}{\neg A, A \vdash A}\ \text{ax} \qquad \dfrac{}{\neg A, A \vdash \neg A}\ \text{ax}}{\neg A, A \vdash \bot}\ \neg_e}{\neg A \vdash A \to \bot}\ \to_i}{\vdash \neg A \to (A \to \bot)}\ \to_i$$

2. Montrer que le séquent $\vdash (A \to \bot) \to \neg A$ est dérivable, en explicitant un arbre de preuve.

   Solution :
   $$\dfrac{\dfrac{\dfrac{\dfrac{}{A \to \bot, A \vdash A}\ \text{ax} \qquad \dfrac{}{A \to \bot, A \vdash A \to \bot}\ \text{ax}}{A \to \bot, A \vdash \bot}\ \to_e}{A \to \bot \vdash \neg A}\ \neg_i}{\vdash (A \to \bot) \to \neg A}\ \to_i$$

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.

   Solution :
   $$\dfrac{\dfrac{}{\vdash \neg A \to (A \to \bot)}\ \text{Q1} \qquad \dfrac{}{\vdash (A \to \bot) \to \neg A}\ \text{Q2}}{\vdash \neg A \to (A \to \bot) \wedge (A \to \bot) \to \neg A}\ \wedge_i$$

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.

   Solution : Attention : il n'est pas demandé un arbre de dérivation mais de montrer que P est vraie pour toute valuation. On peut dessiner la table de vérité de P, où on utilise $A \to B = \neg A \vee B$:

A	B	$\varphi = A \to B$	$\psi = \varphi \to A$	$P = \psi \to A$
0	0	1	0	1
0	1	1	0	1
1	0	0	1	1
1	1	1	1	1

   Remarque : Vu que « $\Gamma \vdash \varphi \implies \Gamma \models \varphi$ » est au programme, on peut aussi démontrer cette question à partir de la suivante.

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

Solution :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, A \vdash A}\,\text{ax} \quad \dfrac{}{\Gamma, A \vdash \neg A}\,\text{ax}}{\Gamma, A \vdash \bot}\,\neg_e}{\Gamma, A \vdash B}\,\bot_e}{\Gamma \vdash A \to B}\,\to_i \qquad \dfrac{}{\Gamma \vdash (A \to B) \to A}\,\text{ax}}{\Gamma \vdash A}\,\to_e$$

II Lois de de Morgan

1. Prouver le séquent $\neg p \vee \neg q \vdash \neg(p \wedge q)$.

   Solution : On note $\Gamma = \{\neg p \vee \neg q, p \wedge q\}$.

$$\dfrac{\dfrac{}{\Gamma \vdash \neg p \vee \neg q}\,\text{ax} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, \neg p \vdash p \wedge q}\,\text{ax}}{\Gamma, \neg p \vdash p}\,\wedge_e \quad \dfrac{}{\Gamma, \neg p \vdash \neg p}\,\text{ax}}{\Gamma, \neg p \vdash \bot}\,\neg_e \quad \dfrac{\dfrac{\dfrac{}{\Gamma, \neg q \vdash p \wedge q}\,\text{ax}}{\Gamma, \neg q \vdash q}\,\wedge_e \quad \dfrac{}{\Gamma, \neg q \vdash \neg q}\,\text{ax}}{\Gamma, \neg q \vdash \bot}\,\neg_e}{\Gamma \vdash \bot}\,\vee_e}{\neg p \vee \neg q \vdash \neg(p \wedge q)}\,\neg_i$$

2. Prouver le séquent $\neg(p \vee q) \vdash \neg p \wedge \neg q$.

   Solution :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\neg(p \vee q), p \vdash \neg(p \vee q)}\,\text{ax} \quad \dfrac{\dfrac{}{\neg(p \vee q), p \vdash p}\,\text{ax}}{\neg(p \vee q), p \vdash p \vee q}\,\vee_i}{\neg(p \vee q), p \vdash \bot}\,\neg_e}{\neg(p \vee q) \vdash \neg p}\,\neg_i \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\neg(p \vee q), q \vdash \neg(p \vee q)}\,\text{ax} \quad \dfrac{\dfrac{}{\neg(p \vee q), q \vdash q}\,\text{ax}}{\neg(p \vee q), q \vdash p \vee q}\,\vee_i}{\neg(p \vee q), q \vdash \bot}\,\neg_e}{\neg(p \vee q) \vdash \neg q}\,\neg_i}{\neg(p \vee q) \vdash \neg p \wedge \neg q}\,\wedge_i$$

3. Prouver le séquent $\neg p \wedge \neg q \vdash \neg(p \vee q)$.

   Solution : On note $\Gamma = \{\neg p \wedge \neg q, p \vee q\}$.

$$\dfrac{\dfrac{}{\Gamma \vdash p \vee q}\,\text{ax} \quad \dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma, p \vdash p}\,\text{ax} \quad \dfrac{\dfrac{}{\Gamma, p \vdash \neg p \wedge \neg q}\,\text{ax}}{\Gamma, p \vdash \neg p}\,\wedge_e}{\Gamma, p \vdash \bot}\,\neg_e \quad \dfrac{\dfrac{}{\Gamma, q \vdash q}\,\text{ax} \quad \dfrac{\dfrac{}{\Gamma, q \vdash \neg p \wedge \neg q}\,\text{ax}}{\Gamma, q \vdash \neg q}\,\wedge_e}{\Gamma, q \vdash \bot}\,\neg_e}{\Gamma \vdash \bot}\,\vee_e}{\neg p \wedge \neg q \vdash \neg(p \vee q)}\,\neg_i$$

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

   Solution : On note $\varphi = \neg(p \wedge q)$.

$$\dfrac{\dfrac{}{\varphi \vdash p \vee \neg p}\,\text{te} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\varphi, p, q \vdash \neg(p \wedge q)}\,\text{ax} \quad \dfrac{\dfrac{}{\varphi, p, q \vdash p}\,\text{ax} \quad \dfrac{}{\varphi, p, q \vdash q}\,\text{ax}}{\varphi, p, q \vdash p \wedge q}\,\wedge_i}{\varphi, p, q \vdash \bot}\,\neg_e}{\varphi, p \vdash \neg q}\,\neg_i}{\varphi, p \vdash \neg p \vee \neg q}\,\vee_i \qquad \dfrac{\dfrac{}{\varphi, \neg p \vdash \neg p}\,\text{ax}}{\varphi, \neg p \vdash \neg p \vee \neg q}\,\vee_i}{\neg(p \wedge q) \vdash \neg p \vee \neg q}\,\vee_e$$

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.

Solution :

1. Par induction :
   - si $A = \top$, alors $|A|_v = \top$ et $\dfrac{}{\Gamma \vdash \top}\ \top_i$ est une preuve ;
   - si $A = \bot$, alors $|A|_v = \neg\bot$. On a alors la preuve $\dfrac{\dfrac{}{\Gamma, \bot \vdash \bot}\ \text{ax}}{\Gamma \vdash \neg\bot}\ \neg_i$;
   - si $A = x_i$, $x_i \in V$, alors on a la preuve $\dfrac{}{\Gamma \vdash |x_i|_v}\ \text{ax}$
   - supposons $\Gamma \vdash |B|_v$ et $A = \neg B$. Si $v(A) = 1$, alors $|B|_v = \neg B = A$, donc $\Gamma \vdash A = |A|_v$. Si $v(A) = 0$, alors $|B|_v = B$, $|A|_v = \neg\neg B$ et on a la preuve :
   $$\dfrac{\dfrac{\dfrac{\Gamma \vdash B}{\Gamma, \neg B \vdash B}\ \text{aff} \quad \dfrac{}{\Gamma, \neg B \vdash \neg B}\ \text{ax}}{\Gamma, \neg B \vdash \bot}\ \neg_e}{\Gamma \vdash \neg\neg B}\ \neg_i$$
   - supposons $\Gamma \vdash |B|_v$, $\Gamma \vdash |C|_v$ et $A = B \to C$. Distinguons les cas :
     - si $v(A) = 0$, alors $v(B) = 1$ et $v(C) = 0$. On a donc $|A|_v = \neg A = \neg(B \to C)$. On a donc la preuve :
     $$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash B}{\Gamma, B \to C \vdash B}\ \text{aff} \quad \dfrac{}{\Gamma, B \to C \vdash B \to C}\ \text{ax}}{\Gamma, B \to C \vdash C}\ \to_e \quad \dfrac{\Gamma \vdash \neg C}{\Gamma, B \to C \vdash \neg C}\ \text{aff}}{\Gamma, B \to C \vdash \bot}\ \neg_e}{\Gamma \vdash \neg(B \to C)}\ \neg_i$$
     - sinon, $v(A) = 1$ et $v(B) = 0$ ou $v(C) = 1$ (ou les deux). On a donc $A = B \to C$. Dans le premier cas, on a la preuve :
     $$\dfrac{\dfrac{\dfrac{\dfrac{\Gamma \vdash \neg B}{\Gamma, B \vdash \neg B}\ \text{aff} \quad \dfrac{}{\Gamma, B \vdash B}\ \text{ax}}{\Gamma, B \vdash \bot}\ \neg_e}{\Gamma, B \vdash C}\ \bot_e}{\Gamma \vdash B \to C}\ \to_i$$
     Dans le deuxième cas, on a la preuve :
     $$\dfrac{\dfrac{\Gamma \vdash C}{\Gamma, B \vdash C}\ \text{aff}}{\Gamma \vdash B \to C}\ \to_i$$

IV Quantificateurs

Montrer les séquents suivants :

1. $\vdash \forall x\, A \to \exists x\, A$.

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

Solution :

1. $$\dfrac{\dfrac{\dfrac{}{\forall x\, A \vdash \forall x\, A}\,\text{ax}}{\forall x\, A \vdash A}\,\forall_e}{\forall x\, A \vdash \exists x\, A}\,\exists_i}{\vdash \forall x\, A \exists x\, A}\,\to_i$$

2. On pose $\Gamma = \{\exists x\, \neg A, \forall x\, A\}$. On a :

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \exists x\, \neg A}\,\text{ax} \qquad \dfrac{\dfrac{\dfrac{}{\Gamma, \neg A \vdash \forall x\, A}\,\text{ax}}{\Gamma, \neg A \vdash A}\,\forall_e \qquad \dfrac{}{\Gamma, \neg A \vdash \neg A}\,\text{ax}}{\Gamma, \neg A \vdash \bot}\,\neg_e}{\Gamma \vdash \bot}\,\exists_e}{\exists x\, \neg A \vdash \neg(\forall x\, A)}\,\neg_i$$

L'utilisation de $\exists_e$ est bien valide, car x n'est pas libre dans Γ.

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 :

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

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

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.

   Solution :

   $$\dfrac{\dfrac{\dfrac{}{\Gamma, x : b \vdash g\ x : a}\,(4) \qquad \dfrac{}{\Gamma, x : b \vdash x : b}\,(4)}{\Gamma, x : b \vdash f\ (g\ x)\ x : a}\,(6)}{\Gamma \vdash \texttt{fun}\ x \to f\ (g\ x)\ x : b \to a}\,(5)$$

2. Quelles analogies peut-on faire entre le typage OCaml et la déduction naturelle ?

   Solution :
   (5) est analogue à l'introduction de l'implication en déduction naturelle.

   (6) est analogue à l'élimination de l'implication en 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.

   Solution : Supposons qu'il existe un tel type τ.
   Comme x est une application de fonction, la seule règle permettant de prouver $\vdash$ `(fun g -> g 1 2) (fun x -> 3)` $: \tau$ est (6) avec $f =$ `fun g -> g 1 2` et $e =$ `fun x -> 3`, ce qui donne $\vdash$ `fun g -> g 1 2` $: \sigma \to \tau$ et $\vdash$ `fun x -> 3` $: \sigma$ pour un certain σ.
   La seule règle applicable pour typer e est (5) et (3), ce qui donne $\sigma = \sigma' \to$ `int` pour un certain σ'.
   f doit alors être de type $(\sigma' \to$ `int`$) \to \tau$.

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 $size(\Pi)$, 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 $\mathcal{I}_0$

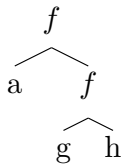

FIGURE 2 – Représentation sous forme d'arbre du terme $f(a, f(g, h))$

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 $\mathcal{I}_0$.

Solution :

$$\dfrac{\dfrac{\dfrac{}{T \vdash f(f(a,k_1),k_2)}\ \textsc{Ax} \quad \dfrac{}{T \vdash k_2}\ \textsc{Ax}}{T \vdash f(a,k_1)}\ \textsc{Red-f} \quad \dfrac{\dfrac{}{T \vdash f(k_1,k_2)}\ \textsc{Ax} \quad \dfrac{}{k_2}\ Ax}{T \vdash k_1}\ \textsc{Red-f}}{T \vdash a}\ \textsc{Ref-f}$$

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. $Termes(\Pi) \subseteq st(T)$.

Solution : On fait une preuve par induction sur l'arbre de preuve.

Si l'arbre de preuve est réduit à la règle Ax alors la preuve est immédiate pour le cas général et le cas particulier.

Sinon, l'arbre de preuve se termine par une autre règle. Nous pouvons faire une étude de cas :

- App-f : par hypothèse d'induction, la propriété que nous souhaitons prouver est vraie.
- Red-f : par minimalité de l'arbre de preuve, nous savons que les sous-arbres Π_1 et Π_2 ne terminent pas par une règle App-f. En effet, si tel était le cas, alors nous pourrions construire un arbre plus petit en omettant les deux dernières étapes. Par conséquent, Π ne contient que des termes issus de $st(T)$.

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 : $size(T, u) = |st(u)| + \Sigma_{t \in T}|st(t)|$.

Solution :
- On calcule tous les sous-termes de $T \cup \{u\}$
- Tant qu'on n'a pas atteint un point fixe, on sature T avec les termes déductibles en une étape (si le terme déduit n'est pas dans $st(T) \cup \{u\}$, alors on ne l'ajoute pas). Le nombre maximal d'itérations est $|st(T) \cup \{u\}|$. On remarquera que $|st(T)|$ est au plus quadratique en la taille du problème car chaque sous terme d'un terme de T est plus petit que ce même terme.
- si u est dans l'ensemble saturé alors on retourne « oui », sinon on retourne « non ».

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.

Solution : Le problème de déduction est clairement dans P avec la question précédente (la taille du problème est le nombre de sous-termes).

L'idée est la suivante :

- nous allons associer à chaque variable propositionnelle x, une constante $v_x \in \mathcal{C}$
- nous allons remarquer que le terme t est déductible depuis $f(\ldots f(f(t, t_1), t_2), \ldots, t_n)$ si et seulement si $t_1, \ldots, t_n$ le sont aussi.

Pour montrer que le problème de déduction est P-complet, on peut utiliser HORN-SAT. Soit $\Phi = C_1 \wedge \ldots \wedge C_n$ une conjonction finie de clauses de Horn. Nous construisons :

1. Pour tout $x \in \mathcal{V}$, on associe une constante $v_x \in \mathcal{C}$. On définit également une constante $v_\perp$.
2. pour tout i, on définit

$$t_i = \begin{cases} f(\ldots f(f(v_x, v_{x_1}), v_{x_2}), \ldots, v_{x_n}) & \text{si } C_i = (x_1 \wedge \ldots \wedge x_n \Rightarrow x) \\ f(\ldots f(f(v_\perp, v_{x_1}), v_{x_2}), \ldots, v_{x_n}) & \text{si } C_i = (x_1 \wedge \ldots \wedge x_n \Rightarrow false) \\ v_x \text{ si } C_i = (true \Rightarrow x) \end{cases}$$

3. Soit $T = \{t_1, \ldots, t_n\}$. Par construction, $T \vdash v_x$ implique $x = true$ (pour tout v_x apparaissant dans Φ) dans toute valuation satisfaisant Φ.
   Par conséquent, $T \vdash v_\perp$ implique Φ est non-satisfiable. Réciproquement, si $T \nvdash v_\perp$ alors nous définissons la valuation $\{x \to 1 \mid T \vdash v_x\} \cup \{x \to 0 \mid T \nvdash v_x\}$ qui satisfait Φ.

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 $u =_{AC} v$.

Solution : Deux remarques ici :

— On peut se restreindre à une terme de la forme $u_1 \oplus ... \oplus u_n$ avec u_i ne contenant pas de $\oplus$.

— On peut simplifier le multi-ensemble obtenu en fonction du nombre d'occurrences de chaque facteur. On garde le facteur si son nombre d'occurrences est impair, on l'enlève s'il est pair. Le facteur 0 peut être retiré du multi-ensemble. On note cette procédure de simplification $Simplify(\cdot)$.

Étant donné que l'on raisonne modulo AC, on peut remarquer que sans perte de généralité, on peut aplatir tous les $\oplus$, i.e., considérer $\oplus$ comme un opérateur n-aire.

Étant donné que u et v sont clos, il suffit de calculer le multi-ensemble des facteurs de u et v en se donnant pour définition : $Facteurs(t) = \begin{cases} \bigcup_{i=1}^{n} Facteurs(t_i) & \text{si } t = t_1 \oplus \ldots \oplus t_n \\ \{t\} & \text{sinon} \end{cases}$

L'algorithme est comme suit :

$$Egal(u, v) = \begin{cases} \perp & \text{si } u = f(t_1, t_2) \text{ and } v \neq f(t'_1, t'_2) \text{ (ou inversement)} \\ Egal(t_1, t'_1) \wedge Egal(t_2, t'_2) & \text{si } u = f(t_1, t_2) \text{ and } v = f(t'_1, t'_2) \\ Simplify(Facteurs(u)) == Simplify(Facteurs(v)) & \text{sinon} \end{cases}$$

Question 6. Soit T un ensemble de termes clos. Soit t un terme clos.

Montrer que $T \vdash t$ dans $\{\text{GX}, \text{Ax'}\}$ est décidable en temps polynomial.

Solution : On peut commencer par remarquer que sans perte de généralité, on peut supposer que notre arbre est de hauteur 2 : une unique application de GX et ensuite uniquement des applications de Ax'. En effet, si on a deux étages, on peut juste les « aplatir » et on obtient toujours un arbre de preuve valide.

On définit $Facteurs(t)$ et $Simplify(S)$ comme dans la réponse à la question précédente.

Voici l'algorithme :

1. On calcule $S_t = Simplify(Facteurs(t))$ et $S_T = Simplify(Facteurs(T))$ (temps polynomial)
2. Si $S_t \not\subseteq S_T$ alors t n'est pas déductible à partir de T. En effet, il existe un sous-terme de t que nous ne pourrons jamais construire à partir de T (temps polynomial)
3. Sinon on représente S_t comme un vecteur de taille $|S_T|$ avec pour valeurs 0/1. Le vecteur a un 1 en position i si le i-ème facteur de S_T est aussi présent dans S_t. Sinon, on affecte la valeur 0. Notons ce vecteur V^t.
4. On peut définir les mêmes vecteurs pour les différents termes de T pris individuellement, notons les V_i^T
5. la déductibilité de t est maintenant réduite à l'existence d'une combinaison linéaire dans $\mathbb{F}_2^p$ des vecteurs V_i^T telle que $\Sigma_i \alpha_i V_i^T = V^t$. Cette étape se calcule en temps polynomial à l'aide d'un pivot de Gauss.