Composition d'option informatique n°6

(Durée : 4 heures)

L'utilisation de la calculatrice **n'est pas autorisée** pour cette épreuve.

⋆ ⋆ ⋆

1 Préliminaires

1.1 Premières preuves

Question 1 On propose :

$$\dfrac{\dfrac{\dfrac{}{\vdash x, x^\perp}\,\text{ax} \quad \dfrac{}{\vdash y, y^\perp}\,\text{ax}}{\vdash x \otimes y^\perp, x^\perp, y}\,\otimes}{\vdash (x \otimes y^\perp) \parr x^\perp, y}\,\parr}{\vdash ((x \otimes y^\perp) \parr x^\perp) \parr y}\,\parr$$

Question 2 On a les deux preuves suivantes. On utilise une notation similaire à une règle d'inférence, mais sans nom, lorsqu'on se contente de réécrire la formule par une formule qui lui est égale syntaxiquement.

– $\vdash ((x \parr y) \parr z) \multimap (x \parr (y \parr z))$ prouvable par :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash x, x^\perp}\,\text{ax} \quad \dfrac{}{\vdash y, y^\perp}\,\text{ax}}{\vdash x^\perp \otimes y^\perp, x, y}\,\otimes \quad \dfrac{}{\vdash z, z^\perp}\,\text{ax}}{\vdash (x^\perp \otimes y^\perp) \otimes z^\perp, x, y, z}\,\otimes}{\vdash ((x^\perp \otimes y^\perp) \otimes z^\perp) \parr (x \parr (y \parr z))}\,\parr \times 3}{\vdash ((x \parr y) \parr z)^\perp \parr (x \parr (y \parr z))}}{\vdash ((x \parr y) \parr z) \multimap (x \parr (y \parr z))}$$

– $\vdash ((x \multimap y) \otimes (y \multimap z)) \multimap (x \multimap z)$ prouvable par :

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash x, x^\perp}\,\text{ax} \quad \dfrac{\dfrac{}{\vdash y, y^\perp}\,\text{ax} \quad \dfrac{}{\vdash z, z^\perp}\,\text{ax}}{\vdash y^\perp, y \otimes z^\perp, z}\,\otimes}{\vdash x \otimes y^\perp, y \otimes z^\perp, x^\perp, z}\,\otimes}{\vdash ((x \otimes y^\perp) \parr (y \otimes z^\perp)) \parr (x^\perp \parr z)}\,\parr \times 3}{\vdash ((x \multimap y) \otimes (y \multimap z)) \multimap (x \multimap z)}$$

Question 3 On raisonne par induction sur A :

– si $A = x \in \mathcal{V}$, alors $\dfrac{}{\vdash x, x^\perp}\,\text{ax}$ est une preuve de $\vdash A, A^\perp$.

– supposons que $\vdash B, B^\perp$ et $\vdash C, C^\perp$ soient prouvables. Distinguons selon la formule A :

* si $A = B \parr C$, alors $\dfrac{\dfrac{\vdash B, B^\perp \quad \vdash C, C^\perp}{\vdash B \parr C, B^\perp \otimes C^\perp}\otimes}{\dfrac{\vdash B \parr C, (B \parr C)^\perp}{\vdash A, A^\perp}}$ est une preuve de $\vdash A, A^\perp$;

* si $A = B \otimes C$, alors $\dfrac{\dfrac{\vdash B, B^\perp \quad \vdash C, C^\perp}{\vdash B \otimes C, B^\perp \parr C^\perp}\otimes}{\dfrac{\vdash B \otimes C, (B \otimes C)^\perp}{\vdash A, A^\perp}}$ est une preuve de $\vdash A, A^\perp$.

On conclut par induction.

Question 4 On a l'arbre de preuve suivant :

$$\dfrac{\vdash \Gamma, A \multimap B \quad \dfrac{\vdash \Delta, A \quad \dfrac{}{\vdash B, B^\perp}\,\text{ax}}{\vdash \Delta, B, A \otimes B^\perp}\otimes}{\vdash \Gamma, \Delta, B}\,\text{cut}$$

Question 5 On montre le résultat par induction sur A :

- si $A = x$, alors $|A^\perp|_x = |x^\perp|_x = 0 = |x|_{x^\perp}$;
- sinon, supposons le résultat établi pour B et C. On distingue :
  * si $A = B \otimes C$, alors $|A^\perp|_x = |B^\perp \parr C^\perp|_x = |B^\perp|_x + |C^\perp|_x = |B|_{x^\perp} + |C|_{x^\perp} = |A|_{x^\perp}$;
  * de même si $A = B \parr C$.

On conclut par induction.

Question 6 Montrons un résultat un peu plus fort que celui demandé par l'énoncé, à savoir que si $\vdash \Gamma$ est prouvable, alors $|\Gamma|_x = |\Gamma|_{x^\perp}$, où $|\Gamma|_x = \sum_{A \in \Gamma} |A|_x$ (de même pour $|\Gamma|_{x^\perp}$). Montrons ce résultat par induction sur l'arbre de preuve de $\vdash \Gamma$. Pour ce faire, montrons que pour toute règle, si les prémisses vérifient l'égalité, alors la conclusion aussi.

- La règle (ax) vérifie l'égalité trivialement.
- Supposons que $|\Gamma, A|_x = |\Gamma, A|_{x^\perp}$ et $|\Delta, A^\perp|_x = |\Delta, A^\perp|_{x^\perp}$. Alors :

  $$|\Gamma, \Delta|_x = |\Gamma, A|_x - |A|_a + |\Delta, A^\perp|_x - |A^\perp|_x = |\Gamma, A|_{x^\perp} - |A^\perp|_{x^\perp} + |\Delta, A^\perp|_{x^\perp} - |A|_{x^\perp} = |\Gamma, \Delta|_{x^\perp}$$

  On a ici utilisé le résultat prouvé précédemment. La règle (cut) vérifie donc la propriété voulue.
- Supposons que $|\Gamma, A|_x = |\Gamma, A|_{x^\perp}$ et $|\Delta, B|_x = |\Delta, B|_{x^\perp}$. Alors :

  $$|\Gamma, A \otimes B, \Delta|_x = |\Gamma, A|_x + |\Delta, B|_x = |\Gamma, A|_{x^\perp} + |\Delta, B|_{x^\perp} = |\Gamma, A \otimes B, \Delta|_{x^\perp}$$

  La règle ($\otimes$) vérifie donc la propriété.
- Supposons que $|\Gamma, A, B|_x = |\Gamma, A, B|_{x^\perp}$. Alors :

  $$|\Gamma, A \parr B|_x = |\Gamma, A, B|_x = |\Gamma, A, B|_{x^\perp} = |\Gamma, A \parr B|_{x^\perp}$$

  La règle ($\parr$) vérifie donc la propriété.

On conclut par induction et en posant $\Gamma = \{A\}$.

Question 7 Si on pose $A = (x \otimes x) \multimap x$, alors $A = (x \otimes x)^\perp \parr x = x^\perp \parr x^\perp \parr x$. On a donc $|A|_x = 1 \neq 2 = |A|_{x^\perp}$. Par contraposée de la question précédente, on en déduit que A n'est pas prouvable.

1.2 Implémentation

Question 8 On se contente de vérifier que les constructeurs sont identiques, et que les sous-formules aussi. On réunit les cas de filtrages avec les mêmes noms de variables.

```
let rec egales a b = match a, b with
    | Var i, Var j | Neg i, Neg j -> i = j
    | Par (a1, b1), Par (a2, b2)
    | Tens (a1, b1), Tens (a2, b2) -> egales a1 a2 && egales b1 b2
    | _ -> false
```

Il y a une vérification en $\mathcal{O}(1)$ pour chaque nœud de l'arbre syntaxique de chaque arbre, et on s'arrête à la première contradiction, soit une complexité en $\mathcal{O}(\min(|A|, |B|))$.

Question 9 À nouveau, pas de grande difficulté.

```
let rec neg = function
    | Var i -> Neg i
    | Neg i -> Var i
    | Par (a, b) -> Tens (neg a, neg b)
    | Tens (a, b) -> Par (neg a, neg b)
```

Question 10 On teste l'égalité pour chaque élément de la liste, jusqu'à en trouver un égal à `a`.

```
exception ErreurPreuve of formule

let rec filtre a = function
    | [] -> raise (ErreurPreuve a)
    | b :: q -> if egales a b then q else b :: filtre a q
```

Question 11 Le cas de l'axiome est direct. Pour les autres règles, on calcule récursivement les séquents prouvés par les sous-preuves, puis on filtre les bonnes formules dans ces séquents pour reformer la nouvelle liste.

```
let rec verification = function
    | Ax i -> [Var i; Neg i]
    | Cut (a, p1, p2) ->
        let seq1 = verification p1 and seq2 = verification p2 in
        filtre a seq1 @ filtre (neg a) seq2
    | PPar (a, b, p) ->
        let seq = verification p in
        Par (a, b) :: filtre a (filtre b seq)
    | PTens (a, b, p1, p2) ->
        let seq1 = verification p1 and seq2 = verification p2 in
        Tens (a, b) :: filtre a seq1 @ filtre b seq2
```

Question 12 Pour chaque règle de la preuve, outre les appels récursifs, on fait des appels à :

- `filtre`, de complexité en $|A| \times |\Gamma|$ si on fait un appel `filtre a gamma` ;
- la concaténation `@`, de complexité linéaire en la taille du premier séquent.

Pour garantir la complexité polynomiale, il s'agit donc de garantir que toute formule apparaissant dans la preuve et tout séquent sont de tailles polynomiales en le nombre de règles. C'est bien le cas, car le nombre total de littéraux (x ou $x^\perp$) et de connecteurs ($\otimes$ et $\parr$) en conclusion d'une preuve est au plus le double du

nombre de règles appliquées (car (ax) fait apparaître deux littéraux, ($\otimes$) et ($\parr$) font apparaître un connecteur, et (cut) en fait disparaître).

2 Réseaux de preuve

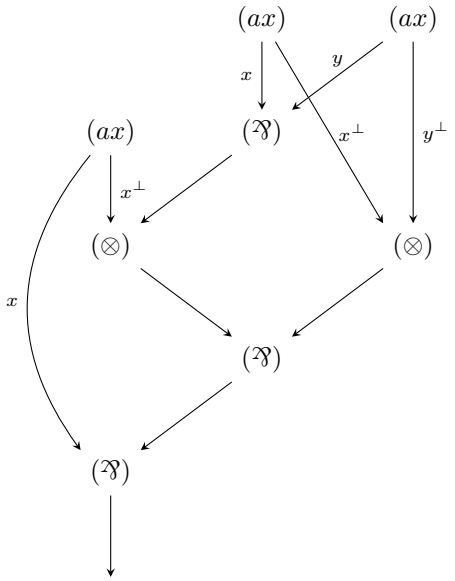
2.1 Réseau

Question 13 On remarque que les seuls sommets qui ont à la fois des prémisses et conclusions sont les sommets ($\parr$) et ($\otimes$). Pour chacun de ces sommets, on remarque que si A et B sont les étiquettes des prémisses et C l'étiquette de la conclusion, alors $|C| > |A|$ et $|C| > |B|$ (où $|A|$ est la taille de la formule, c'est-à-dire le nombre de symboles).

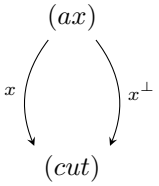
S'il existe un cycle orienté, ce cycle est de la forme $s_0 \xrightarrow{A_1} s_1 \xrightarrow{A_2} \dots \xrightarrow{A_n} s_n$, avec $s_n = s_0$. Par ce qui a été dit plus haut, on a, pour $i \in [\![1, n-1]\!]$, $|A_i| < |A_{i+1}|$. De même, $|A_n| < |A_1|$. Mais alors, par transitivité, $|A_1| < |A_1|$, ce qui est absurde.

2.2 Réseaux de preuve

Question 14 On obtient le réseau de preuve suivant. Attention à ne pas inverser les deux arêtes x (si on les inverse, l'arbre de preuve associé est incorrect).



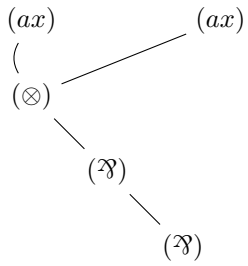
Question 15 Le réseau suivant n'est pas un réseau de preuve :



En effet, en accord avec la construction des réseaux de preuve, cela correspondrait à un arbre de preuve de la forme $\dfrac{\vdash x \quad \vdash x^\perp}{\vdash}$ cut qui est évidemment incorrect ($\vdash x$ et $\vdash x^\perp$ ne sont pas prouvables d'après la question 6).

2.3 Critère de Danos et Régnier

Question 16 Le graphe suivant est un interrupteur :



Question 17 Un sommet ($\parr$) ayant deux prémisses, on en déduit directement que le nombre d'interrupteurs est 2^k.

Question 18 Par induction sur les arbres de preuve :

- pour un axiome $\dfrac{}{\vdash x, x^\perp}\ \text{ax}$, le seul interrupteur est un sommet (ax) seul. C'est effectivement un graphe connexe sans cycle.
- pour une règle $\dfrac{\vdash \Gamma, A \quad \vdash \Delta, B}{\vdash \Gamma, A \otimes B, \Delta}\ \otimes$, supposons que les réseaux $\mathcal{R}$ et $\mathcal{R}'$ de conclusions Γ, A et Δ, B ont des interrupteurs connexes sans cycle. Alors un interrupteur associé au réseau de conclusions $\Gamma, A \otimes B, \Delta$ s'obtient en rajoutant un sommet $(\otimes)$ adjacent à un sommet d'un interrupteur de $\mathcal{R}$ et à un sommet d'un interrupteur de $\mathcal{R}'$. Cela forme bien un graphe connexe sans cycle : le nouveau graphe est connexe, et le nombre d'arêtes est bien égal au nombre de sommets -1 (on avait une différence de 1 dans chaque interrupteur, et on rajoute un sommet et deux arêtes).
- pour une règle $\dfrac{\vdash \Gamma, A, B}{\vdash \Gamma, A \parr B}\ \parr$, supposons que le réseau $\mathcal{R}$ de conclusions Γ, A, B ait des interrupteurs connexes sans cycle. Alors un interrupteur associé au réseau de conclusions $\Gamma, A \parr B$ s'obtient en ajoutant un sommet $(\parr)$ adjacent à sommet d'un interrupteur de $\mathcal{R}$. Cela donne bien un graphe connexe sans cycle : le nouveau graphe est connexe, et on a ajouté une arête et un sommet.

On conclut par induction.

2.4 Un théorème d'élimination des coupures

Question 19 On reprend la preuve du sens direct du théorème et on traite les sommets (cut) comme les sommets $(\otimes)$.

Question 20 On sait qu'un sommet (cut), dans le cas général, a une prémisse A et une prémisse $A^\perp$. Ainsi, si l'un des deux côté est un sommet $(\otimes)$, de conclusion $A \otimes B$, alors l'autre côté est nécessairement de conclusion $(A \otimes B)^\perp = A^\perp \parr B^\perp$, qui est bien conclusion d'un sommet $(\parr)$.

Question 21 Dans un premier temps, remarquons que l'élimination d'une coupure reliée à un sommet (ax) ne brise ni la connexité, ni l'acyclicité. On considère une élimination d'une coupure $\otimes/\parr$ dans un réseau $\mathcal{R}$ qui vérifie le critère de Danos et Régnier. On suppose que $a(A)$ et $b(B)$ sont les arêtes prémisses du sommet $(\otimes)$, et que $c(A^\perp)$ et $d(B^\perp)$ sont les arêtes prémisses du sommet $(\parr)$.

Soit I un interrupteur du réseau $\mathcal{R}'$ obtenu après élimination de la coupure.

- I est acyclique : s'il existe un cycle dans I, s'il passe uniquement par la coupure $a-c$ (resp. $b-d$), alors dans l'interrupteur dans $\mathcal{R}$ où on déconnecte d (resp. c) du sommet ($\parr$), ce cycle existe dans $\mathcal{R}$, ce qui est absurde. Sinon, c'est un cycle de toute façon dans tout interrupteur de $\mathcal{R}$;
- I est connexe : soit J l'interrupteur de $\mathcal{R}$ obtenu à partir de I où on déconnecte c du sommet ($\parr$). Par connexité, toute arête est reliée à a, b, c ou d. Dans I, il y a donc au plus deux composantes connexes : celle reliée à $a-c$ et celle à $b-d$. Cependant, s'il existe un autre chemin de c à a ou b que l'arête $a-c$, il existerait un cycle. c est donc reliée à b via l'arête $c-d$, donc le graphe est connexe.

On conclut par récurrence.

Question 22 L'algorithme termine car le nombre de sommets du réseau est un variant : il diminue strictement (de 1 ou 2) à chaque élimination de coupure. Par construction, une coupure est toujours reliée soit à un sommet (ax), soit à un sommet ($\otimes$) (resp. ($\parr$)) et donc un sommet ($\parr$) (resp. ($\otimes$)), soit à un sommet (cut).

Dans ce dernier cas, on peut toujours appliquer l'une des deux règles au sommet (cut) qui apparaît en premier dans un ordre topologique du réseau (qui existe car un réseau est sans cycle). Lorsque l'algorithme termine, il n'y a donc plus de coupure.

Question 23 On remarque que s'il est appliqué à un réseau, l'algorithme d'élimination des coupures ne change pas les arêtes conclusions du réseau.

Ainsi, s'il existe une preuve d'un séquent $\vdash \Gamma$, on considère le réseau de preuve associé. Par la question 19, il vérifie le critère de Danos et Régnier. En appliquant l'algorithem d'élimination des coupures, on obtient un réseau vérifiant le critère de Danos et Régnier, ne contenant pas de sommet (cut), et ayant les mêmes conclusions. Par le théorème de Danos et Régnier, c'est un réseau de preuve. La preuve correspondante est bien une preuve du séquent $\vdash \Gamma$ n'utilisant pas la coupure, qui est donc admissible.

3 Test de connexité

Question 24 On écrit un algorithme de parcours en profondeur avec une fonction auxiliaire récursive. Le tableau `vus` garde en mémoire les sommets vus, et le pointeur `nb` le nombre de sommets vus.

```
void connexe_rec(graphe g, bool* vus, int s, int* nb){
    if (vus[s]) return;
    vus[s] = true;
    (*nb)++;
    for (int i=0; i<g.deg[s]; i++){
        connexe_rec(g, vus, g.adj[s][i], nb);
    }
}
```

Une fois cette fonction écrite, on obtient la connexité en lançant un parcours depuis le sommet 0 (par exemple) :

```
bool connexe(graphe g){
    bool* vus = malloc(g.n * sizeof(bool));
    for (int s=0; s<g.n; s++) vus[s] = false;
    int nb = 0;
    connexe_rec(g, vus, 0, &nb);
    free(vus);
    return nb == g.n;
}
```

Question 25 Plutôt que d'écrire une fonction qui teste l'acyclicité, on se contente de compter le nombre

d'arêtes (ça tombe bien, on a accès au degré, donc on peut faire ça en $\mathcal{O}(|\mathcal{S}|)$), égal à la somme des degrés divisé par 2.

```
int arbre(graphe g){
    int sd = 0;
    for (int s=0; s<g.n; s++) sd += g.deg[s];
    return g.n == 1 + sd / 2 && connexe(g);
}
```

Question 26 La fonction connexe a une complexité temporelle en $\mathcal{O}(|\mathcal{S}| + |\mathcal{A}|)$. La complexité spatiale correspond au tableau `vus` et à la pile d'appels récursifs. Dans les deux cas on est en $\mathcal{O}(|\mathcal{S}|)$.

Dans la fonction arbre, on n'augmente pas la complexité spatiale par rapport à `connexe`, et on ajoute un terme $\mathcal{O}(|\mathcal{S}|)$ à la complexité temporelle. Toutefois, comme on fait la vérification du nombre d'arêtes avant l'appel à `connexe`, on a la garantie que $|\mathcal{A}| = \mathcal{O}(|\mathcal{S}|)$.

En conclusion : temporel et spatial en $\mathcal{O}(|\mathcal{S}|)$.

Question 27 On se contente de garder en mémoire le sommet courant et le nombre d'itérations. On évite une fonction récursive pour garantir la complexité spatiale constante (seulement deux variables).

```
bool chemin_alea(graphe g, int s, int t, int etapes){
    int u = s;
    int compt = 0;
    while (u != t && compt < etapes){
        compt++;
        u = g.adj[u][randint(g.deg[u])];
    }
    return u == t;
}
```

Question 28 C'est une simple application de la formule des espérances totales. On peut prouver ce résultat sans ça, toutefois :

$$\begin{aligned}\mathbb{E}(X_s) &= \sum_{k\in\mathbb{N}} k\mathbb{P}(X_s = k)\\ &= 1 + \sum_{k\in\mathbb{N}^*} (k-1)\mathbb{P}(X_s = k)\end{aligned}$$

Or $\mathbb{P}(X_s = k) = \sum_{u \text{ voisin de } s} \frac{1}{\deg(s)}\mathbb{P}(X_u = k-1)$ d'après la formule des probabilités totales : le premier sommet atteint par la marche est un voisin de s, et ils sont tous équiprobables. La marche sera de taille k si la marche à partir de ce voisin est de taille $k-1$. En réinjectant, et en intervertissant les sommes, on obtient :

$$\begin{aligned}\mathbb{E}(X_s) &= 1 + \sum_{k\in\mathbb{N}^*} (k-1) \sum_{u \text{ voisin de } s} \frac{1}{\deg(s)}\mathbb{P}(X_u = k-1)\\ &= 1 + \frac{1}{\deg(s)} \sum_{u \text{ voisin de } s} \sum_{k\in\mathbb{N}^*} (k-1)\mathbb{P}(X_u = k-1)\\ &= 1 + \frac{1}{\deg(s)} \sum_{u \text{ voisin de } s} \mathbb{E}(X_u)\end{aligned}$$

Ce qui correspond au résultat attendu.

Question 29 On se contente de lancer une marche aléatoire de taille $2n^3$, où $n = |\mathcal{S}|$.

```
bool connectes_alea(graphe g, int s, int t){
    return chemin_alea(g, s, t, 2 * g.n * g.n * g.n);
}
```

En effet :

- si s et t ne sont pas dans la même composante connexe, alors la fonction renverra toujours `false`, donc il n'y a pas de faux positif ;
- il y a un faux négatif si la marche n'a pas atteint t au bout de $2n^3$ étapes. Or, d'après l'inégalité de Markov, $\mathbb{P}(X_s \geqslant 2n^3) \leqslant \frac{\mathbb{E}(X_s)}{2n^3} \leqslant \frac{1}{2}$.

Enfin, la complexité spatiale est constante, comme pour la fonction `chemin_alea`, et la complexité temporelle est donnée par le nombre d'itérations maximal, c'est-à-dire $\mathcal{O}(n^3)$, ce qui est bien polynomial.

4 Logique directe

Question 30 Soit $\vdash \Gamma$ un séquent prouvable en logique linéaire multiplicative. Par le théorème d'élimination des coupures, il existe une preuve sans coupure. Dans cette preuve, chaque utilisation d'une règle augmente la taille de 2 (ax) ou 1 ($\otimes$ et $\parr$) du séquent conclusion. On en déduit qu'une telle preuve est de taille polynomiale en la taille du séquent. On peut vérifier en temps polynomial (questions 11 et 12) que cette preuve est correcte et prouve bien $\vdash \Gamma$.

Ceci montre bien que $\texttt{LLM} \in \mathsf{NP}$.

Question 31 On suit la définition. On obtient : $\vdash x \parr x, (x^\perp \otimes s_0^\perp) \parr (x^\perp \otimes (s_1^\perp \parr s_1^\perp)) \parr (x^\perp \otimes (s_2^\perp \parr s_2^\perp) \parr (x^\perp \otimes s_3^\perp), (s_0 \parr s_1) \otimes (s_1 \parr s_2) \otimes (s_2 \parr s_3)$.

Question 32 Pour chaque arête $a = \{s, t\}$, on sait que $s \in X$ ou $t \in X$. Sans perte de généralité, supposons $s \in X$. Alors la preuve suivante est une preuve de $\vdash s^\perp, s \parr t$:

$$\dfrac{\dfrac{\dfrac{}{\vdash s, s^\perp}\,\text{ax}}{\vdash s^\perp, s, t}\,\text{aff}}{\vdash s^\perp, s \parr t}\,\parr$$

En combinant les preuves pour $a = \{s, t\}$ et $a' = \{s', t'\}$ avec la règle $\otimes$, on obtient une preuve :

$$\dfrac{\vdash s^\perp, s \parr t \qquad \vdash s'^\perp, s' \parr t'}{\vdash s^\perp, s'^\perp, (s \parr t) \otimes (s' \parr t')}\,\otimes$$

En combinant ces preuves pour chaque arête de $\mathcal{A}$, on obtient une preuve de $\vdash \Delta', B(G)$, où Δ' est un sous-ensemble de Δ. En effet, pour chaque sommet $s \in X$, le nombre d'occurences de $s^\perp$ est inférieur ou égal à son degré (car le sommet s n'est choisi qu'au plus une fois par arête incidente).

On peut alors utiliser la règle d'affaiblissement pour avoir une preuve de $\vdash \Delta, B(G)$.

Question 33 En utilisant la règle $\parr$ autant de fois que nécessaire sur le séquent de la question précédente, si $X = \{s_1, s_2, \ldots, s_k\}$, on obtient une preuve de $\vdash (s_1^\perp)^{\deg(s_1)}, (s_2^\perp)^{\deg(s_2)}, \ldots, (s_k^\perp)^{\deg(s_k)}, B(G)$. En utilisant k fois la règle $\dfrac{}{\vdash x, x^\perp}\,\text{ax}$, puis k fois la règle $\otimes$, on obtient une preuve de $\vdash \underbrace{x, x, \ldots, x}_{k \text{ fois}}, (x^\perp \otimes (s_1^\perp)^{\deg(s_1)}), \ldots, (x^\perp \otimes (s_k^\perp)^{\deg(s_k)}), B(G)$.

En appliquant la règle d'affaiblissement $|\mathcal{S}| - k$ fois (pour faire apparaître les $(x^\perp \otimes (s^\perp)^{\deg(s)})$, pour $s \notin X$), puis $k + |\mathcal{S}| - 2$ fois la règle $\parr$, on obtient bien une preuve de $\Gamma(G, k)$.

Question 34 Il y a une induction cachée dans la preuve : on suppose qu'on applique inductivement les constructions décrites aux sous-preuves pour obtenir le résultat.

(a) On suppose qu'il existe une formule $A \otimes B$ tel que dans la preuve de $\vdash \Gamma$, A ou B est affaiblie. Sans perte de généralité, supposons que A est affaiblie. Alors l'apparition de la formule $A \otimes B$ se fait par l'application d'une règle de la forme :

$$\frac{\vdash \Lambda, A \qquad \vdash \Delta, B}{\vdash \Lambda, A \otimes B, \Delta} \otimes$$

où $\vdash \Lambda, A$ et $\vdash \Delta, B$ sont prouvables. Si on considère P une preuve de $\vdash \Lambda, A$ où A est affaiblie, alors en supprimant l'application de la règle (aff), on obtient une preuve P' de $\vdash \Lambda$. Alors la preuve suivante est une preuve où ni A, ni B n'a été affaiblie :

$$\frac{\dfrac{P'}{\vdash \Lambda}}{\vdash \Lambda, A \otimes B, \Delta} \text{ aff}$$

(b) On suppose qu'il existe une formule $A \parr B$ tel que dans la preuve de $\vdash \Gamma$, A et B sont affaiblies.Alors l'apparition de la formule $A \parr B$ se fait par l'application d'une règle de la forme :

$$\frac{\vdash \Delta, A, B}{\vdash \Delta, A \parr B} \otimes$$

où $\vdash \Delta, A, B$ sont prouvables. Si on considère P une preuve de $\vdash \Delta, A, B$ où A et B sont affaiblies, alors en supprimant l'application des règles (aff), on obtient une preuve P' de $\vdash \Delta$. Alors la preuve suivante est une preuve où ni A, ni B n'a été affaiblie :

$$\frac{\dfrac{P'}{\vdash \Delta}}{\vdash \Delta, A \parr B} \text{ aff}$$

Question 35 Si X n'est pas une couverture des arêtes par les sommets, alors il existe une arête $a = \{s, t\}$ telle que $s \notin X$ et $t \notin X$. D'après la question précédente, il existe une preuve de $\Gamma(G, k)$ où $(s \parr t)$ n'est pas affaiblie (car elle apparaît avec $\otimes$ dans la formule $B(G)$), et soit s soit t n'est pas affaibli (car il apparaît dans $s \parr t$). Comme ni s ni t n'apparaît comme axiome, on en déduit que $B(G)$ est affaiblie.

Il existe donc une preuve de $\vdash x^k, A(G)$. Toutefois, comme ce séquent ne contient aucune occurrence d'une formule s, pour $s \in \mathcal{S}$, toujours par la question précédente, on en déduit que $A(G)$ est affaiblie, soit que $\vdash x^k$ est prouvable. C'est absurde, car cela implique que $\vdash x$ est prouvable, ce qui est contradictoire (aucune règle ne peut être appliquée).

Question 36 Un littéral $s^\perp$ n'intervient que dans une sous-formule de $A(G)$. On en déduit que si $|X| = \ell > k$, alors il existe au moins ℓ sous-formules de la forme $x^\perp \otimes (s^\perp)^{\deg(s)}$ qui ne sont pas affaiblies. Toujours par la question 34, cela implique qu'il existe au moins ℓ occurrences de $x^\perp$ qui ne sont pas affaiblies, donc qu'il y a eu ℓ applications de la règle $\dfrac{}{\vdash x, x^\perp}$ ax. C'est absurde, car $\vdash \Gamma(G, k)$ ne contient que $k < \ell$ occurrences du littéral x.

Question 37 Les questions 33, 35 et 36 montre qu'il existe une couverture de G de cardinal k si et seulement si $\vdash \Gamma(G, k)$ est prouvable. Dans le sens réciproque, on pourrait avoir $|X| < k$, mais quitte à rajouter des sommets, il existe une couverture de cardinal k (en supposant que $k \leqslant |\mathcal{S}|$).

Par ailleurs, le séquent $\vdash \Gamma(G, k)$ est de taille polynomiale en la taille du graphe si on suppose $k \leqslant |\mathcal{S}|$ (si ce n'est pas le cas, on peut poser $\vdash \Gamma(G, k) = \vdash$, le séquent vide, qui n'est pas prouvable).

On en déduit que $\mathtt{CPS} \leqslant^{\mathsf{p}}_{\mathsf{m}} \mathtt{LD}$, donc $\mathtt{LD}$ est NP-difficile. Comme $\mathtt{LD} \in \mathsf{NP}$, on en déduit qu'il est NP-complet.

$\star\star\star$