

SESSION 2006

Filière MP (groupes MPI et I)

Épreuve commune aux ENS de Paris, Lyon et Cachan

Filière PC (groupe I)

Épreuve commune aux ENS de Paris et Lyon

INFORMATIQUE

Durée : 4 heures

L'usage de calculatrice est interdit

Ordre topologique et circuits booléens

Le sujet aborde la question du calcul des *fonctions booléennes*, en utilisant le modèle des *circuits booléens*.

La première partie introduit une notion fondamentale (ordre topologique sur un graphe sans cycle) et demande d'écrire ou d'étudier quelques algorithmes de base relatifs aux ordres topologiques. La seconde partie introduit les fonctions et les circuits booléens et aborde plusieurs questions relatives au calcul dans ce modèle, notamment d'expressivité. La troisième partie demande d'estimer des bornes de complexité, supérieures et inférieures.

Les seconde et troisième parties peuvent largement être abordées même si la première n'a pas été complètement résolue.

Définitions et conventions

Graphes. Un *graphe* (orienté) est un couple $G = (S, A)$ où S est un ensemble fini (les *sommets* de G) et A un sous-ensemble de $S \times S$ (les *arêtes* de G). Si G est un graphe à n sommets, on identifiera S à $\{1, \dots, n\}$. Si X est un ensemble, $|X|$ désignera son cardinal.

Si $a = (u, v)$ est une arête, on appelle u son *origine* et on la note $\text{or}(a)$; on appelle v l'*extrémité* de a et on la note $\text{ex}(a)$. Le sommet v est dit *fil*s de u .

Un *chemin* de longueur $s \geq 1$ est une suite d'arêtes $w = a_1, \dots, a_s$ telle que $\text{ex}(a_i) = \text{or}(a_{i+1})$ pour $1 \leq i \leq s-1$. On note $S(w)$ l'ensemble de sommets $\{\text{or}(a_i), 1 \leq i \leq s\} \cup \{\text{ex}(a_s)\}$. Ce chemin est un *cycle* si $\text{ex}(a_s) = \text{or}(a_1)$.

Si j est un sommet de G , on appelle *degré entrant* de j et on note $\text{in}(j)$ le nombre d'arêtes d'extrémité j ; on appelle *degré sortant* de j le nombre d'arêtes d'origine j . Un sommet de degré entrant nul est appelé une *entrée*; un sommet de degré sortant nul est appelé une *sortie*; un sommet qui n'est pas une entrée est dit *interne*.

Structures de données et algorithmes. Dans la suite on manipulera des listes d'entiers, des tableaux d'entiers, de listes d'entiers, de booléens, ... Les indices d'un tableau de taille n vont de 1 à n ; la taille d'un tableau t sera notée $\text{taille}(t)$.

La liste vide est notée *nil*. Une liste L non vide est formée d'un premier élément, noté $\text{tête}(L)$, et de la liste formée par les autres éléments, notée $\text{queue}(L)$. La primitive $\text{concat}(a, L)$ renvoie la liste formée en ajoutant a en tête de L . On note $x \in L$ pour indiquer que l'élément x appartient à la liste L .

Vous pouvez utiliser le langage ou pseudo-langage de votre choix pour l'écriture de fonctions, en utilisant les structures de contrôle usuelles (Pour, Si, Tant que, ...). La question 1.4. de la première partie illustre un choix possible. Certaines questions demandent de donner les principes d'un algorithme; pour ces questions, on ne demande pas d'écrire du pseudo-code, mais de décrire l'algorithme en français.

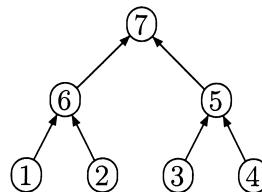
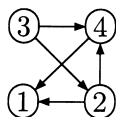
La complexité d'un algorithme désigne le nombre d'opérations élémentaires qu'il effectue : lecture ou écriture dans un tableau, tests, accès à la tête ou à la queue d'une liste, ... La création d'un tableau de taille n a un coût $O(n)$. Si m est une liste, le coût de l'affectation $\ell \leftarrow m$ est 1. On ne cherchera pas à estimer les complexités exactement; on se contentera de donner un ordre de grandeur, en utilisant la notation asymptotique $O(\dots)$.

1 Ordre topologique et applications

Soit $G = (S, A)$ un graphe à n sommets. Un *ordre topologique* sur G est une bijection $o : S = \{1, \dots, n\} \rightarrow \{1, \dots, n\}$ telle que pour toute arête (u, v) de G , $o(u) < o(v)$; l'ordre o est représenté par le tableau $[o[1], \dots, o[n]]$. L'objectif de cette partie est de donner un algorithme pour déterminer un ordre topologique et quelques-unes de ses applications.

Question 1.1.

1. Donner un ordre topologique pour chacun des graphes suivants, quand c'est possible.



2. Montrer que G admet un ordre topologique si et seulement s'il est sans cycle.
3. Écrire une fonction `Inverse(o)` qui calcule le tableau $[o^{-1}(1), \dots, o^{-1}(n)]$. Quelle est sa complexité ?

Le graphe G est donné par *listes d'adjacence*, c'est-à-dire par un tableau t , de taille $|S|$, de listes d'entiers, tel que $j \in t[i]$ si et seulement si (i, j) est une arête de G . De plus, chaque liste est sans répétition.

Question 1.2.

1. Écrire une fonction `Parents(t)` qui renvoie un tableau u , de taille $|S|$, contenant des listes d'entiers, tel que $j \in u[i]$ si et seulement si (j, i) est une arête de G (chaque liste sera sans répétition). Quelle est sa complexité ?
2. Donner les principes d'un algorithme qui renvoie le tableau de taille $|S|$ donnant le degré entrant de chaque sommet et d'un algorithme qui renvoie la liste des entrées de G . Quelle est la complexité de ces algorithmes ?
3. En utilisant les algorithmes précédents, donner les principes d'un algorithme qui calcule un ordre topologique sur G (s'il existe), en complexité $O(|S| + |A|)$.

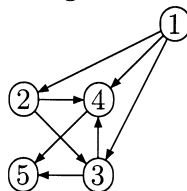
Dans toute la fin de cette partie, on suppose G sans cycle. Si o est un ordre topologique sur G , on dit qu'une liste L d'éléments distincts de $\{1, \dots, n\}$ est *triée selon l'ordre o* si, soit L est vide, soit L est réduite à un élément, soit $o(\text{tête}(L)) < o(\text{tête}(\text{queue}(L)))$ et $\text{queue}(L)$ est triée selon l'ordre o .

Question 1.3. Soit o un ordre topologique sur G . Écrire une fonction `TriSuccesseurs(o, t)`, de complexité $O(|S| + |A|)$, qui trie les listes d'adjacence contenues dans le tableau t selon l'ordre o .

La *fermeture réflexive transitive* de G est le graphe $G^* = (S, A^*)$ tel que (u, v) appartient à A^* si et seulement s'il existe un chemin $a_1, \dots, a_s$ dans G tel que $u = \text{or}(a_1)$ et $v = \text{ex}(a_s)$, ou bien si $u = v$. L'objectif de la fin de cette partie est de calculer cette fermeture. Au vu des questions précédentes, quitte à renuméroter les sommets, on suppose désormais que l'application $o(i) = i$ forme un ordre topologique sur G et que les listes d'adjacence du tableau t sont données triées selon cet ordre.

Question 1.4. On étudie dans cette question l'algorithme de la figure 1, page 4. Il prend en entrée le tableau t des listes d'adjacence triées.

1. Pour le graphe suivant, donner les valeurs de R et Q à chaque entrée dans la boucle des lignes 2—27, ainsi qu'à la fin de l'algorithme.



2. Montrer qu'à l'issue de cet algorithme, le tableau R fournit une description de G^* par listes d'adjacence.

1. $Q \leftarrow \text{Cr erTableau}(\text{faux}, n)$, $R \leftarrow \text{Cr erTableau}(\text{nil}, n)$
 (Q et R sont des tableaux, respectivement de bool ens et de listes d'entiers, respectivement initialis s   faux et nil, et de taille $n = |S|$)
2. Pour i allant de n   1 Faire
 3. $R[i] \leftarrow \text{concat}(i, \text{nil})$
 4. $Q[i] \leftarrow \text{vrai}$
 5. $\ell \leftarrow t[i]$
 6. Tant que $\ell \neq \text{nil}$ Faire
 7. $j \leftarrow \text{t te}(\ell)$
 8. $\ell \leftarrow \text{queue}(\ell)$
 9. Si $Q[j] = \text{faux}$ Faire
 10. $m \leftarrow R[j]$
 11. Tant que $m \neq \text{nil}$ Faire
 12. $k \leftarrow \text{t te}(m)$
 13. $m \leftarrow \text{queue}(m)$
 14. Si $Q[k] = \text{faux}$ Faire
 15. $Q[k] \leftarrow \text{vrai}$
 16. $R[i] \leftarrow \text{concat}(k, R[i])$
 17. Fin Si
 18. Fin Tant que
 19. Fin Si
 20. Fin Tant que
 21. $m \leftarrow R[i]$
 22. Tant que $m \neq \text{nil}$ Faire
 23. $k \leftarrow \text{t te}(m)$
 24. $m \leftarrow \text{queue}(m)$
 25. $Q[k] = \text{faux}$
 26. Fin Tant que
27. Fin Pour
28. Renvoyer R

FIG. 1 – Algorithme de la question 1.4.

3. Soit A^- le sous-ensemble des arêtes de G tel que $(u, v) \in A^-$ si et seulement si $(u, v) \in A$ et s'il n'existe pas de chemin de longueur supérieure ou égale à 2 entre u et v dans G . Montrer que le nombre de sommets internes de G est borné par $|A^-|$, que $|A^*|$ est borné par $|S||A^-| + |S|$ et estimer la complexité de l'algorithme étudié en fonction de $|S|$ et $|A^-|$.

Indication : on pourra montrer que les lignes 10 à 18 sont exécutées si et seulement si (i, j) est dans A^- .

4. La grandeur $|A^-|$ est toujours inférieure à $|S|^2$. Est-il possible de la majorer par une fonction de type $|S|^\alpha$, pour un $\alpha < 2$?

Une *chaîne* est un ensemble de sommets de G soit réduit à un élément, soit de la forme $S(w)$, pour un chemin w .

Question 1.5. On étudie pour conclure cette partie un second algorithme de fermeture réflexive transitive.

1. Écrire une fonction `Chaines` qui calcule une *partition* de S en chaînes, avec la propriété (de minimalité) suivante : pour toutes chaînes S_1 et S_2 calculées par cette fonction, $S_1 \cup S_2$ n'est pas une chaîne.

Cette fonction renverra un tableau de listes d'entiers triées en ordre *croissant* (c'est-à-dire selon l'ordre topologique $o(i) = i$) et sans répétition. Estimer sa complexité.

2. Soit $S_1, \dots, S_k$ une partition de S en chaînes, donnée par un tableau de listes d'entiers sans répétition, triées en ordre croissant. Pour $i = 1, \dots, |S|$ et $h = 1, \dots, k$, on définit $R(h, i) = \min\{j \in S_h \mid (i, j) \in A^*\}$ si cet ensemble n'est pas vide, et $R(h, i) = |S| + 1$ sinon. Enfin, on note C le tableau de taille $|S|$ tel que $C[i]$ est l'unique indice h pour lequel $i \in S_h$.

(a) Montrer que $R(h, i) = i$ si $i \in S_h$ et que, sinon, $R(h, i) = \min\{R(h, j) \mid (i, j) \in A^-\}$ si cet ensemble n'est pas vide.

(b) Soit $(i, j) \in A$ et soit $B = \{\ell \in S \mid \ell < j \text{ et } (i, \ell) \in A^-\}$. Si B est vide, on pose $r = |S| + 1$; sinon, on pose $r = \min\{R(C[j], \ell) \mid \ell \in B\}$. Montrer que $(i, j) \in A^-$ si et seulement si $j < r$.

(c) Écrire une fonction calculant les valeurs $R(h, i)$ en complexité $O(|A| + k(|S| + |A^-|))$.

3. Donner les principes d'un second algorithme qui calcule la fermeture réflexive transitive de G et estimer sa complexité. La sortie sera présentée sous la forme d'un tableau de listes d'adjacence. On ne demande pas que ces listes soient triées.

Remarque : si le graphe G est aléatoire, au sens où chaque arête (i, j) est présente avec une probabilité $0 < \varepsilon < 1$ fixée, ces idées mènent à un algorithme de complexité moyenne $O(|S|^2 \log(|S|))$, contre $O(|S|^{2.5})$ pour celui de la question 1.4.

2 Fonctions booléennes et circuits booléens

Définitions. Pour $n \geq 1$, on note Γ_n l'ensemble des fonctions $\{0, 1\}^n \rightarrow \{0, 1\}$ et on pose $\Gamma = \cup_{n \geq 1} \Gamma_n$; Γ est l'ensemble des *fonctions booléennes*. On note $\vee \in \Gamma_2$ la fonction *ou*, $\wedge \in \Gamma_2$ la fonction *et*, $\oplus \in \Gamma_2$ la fonction *ou exclusif* et $\neg \in \Gamma_1$ la fonction *non*, définies ci-dessous :

x_1	x_2	$x_1 \vee x_2$	x_1	x_2	$x_1 \wedge x_2$	x_1	x_2	$x_1 \oplus x_2$	x_1	$\neg x_1$
0	0	0	0	0	0	0	0	0	0	1
0	1	1	0	1	0	0	1	1	1	0
1	0	1	1	0	0	1	0	1	0	1
1	1	1	1	1	1	1	1	0	1	0

On définit également les fonctions constantes $\mathbf{0}$ et $\mathbf{1} \in \Gamma_1$ qui valent respectivement 0 et 1.

Pour $n \geq 1$, $x = (x_1, \dots, x_n)$ et $x' = (x'_1, \dots, x'_n)$ dans $\{0, 1\}^n$, on note $x \leq x'$ si $x_i \leq x'_i$ pour tout i . Pour un sous-ensemble non vide $S = \{s_1, \dots, s_N\}$ de $\{1, \dots, n\}$, on note $\oplus_{i \in S} x_i$ la valeur $x_{s_1} \oplus \dots \oplus x_{s_N}$ (cette valeur est bien définie, en vertu de l'associativité et de la commutativité de $\oplus$); si S est vide, $\oplus_{i \in S} x_i$ vaut 0. Finalement, une fonction $f \in \Gamma_n$ est dite :

- *monotone* si $x \leq x'$ implique $f(x) \leq f(x')$ pour tous x, x' dans $\{0, 1\}^n$;
- *auto-adjointe* si $f(\neg x_1, \dots, \neg x_n) = \neg f(x_1, \dots, x_n)$ pour tout $(x_1, \dots, x_n)$ dans $\{0, 1\}^n$;
- *affine* s'il existe $S \subset \{1, \dots, n\}$ et $c \in \{0, 1\}$ tels que $f(x) = c \oplus (\oplus_{i \in S} x_i)$ pour tout $x = (x_1, \dots, x_n)$ dans $\{0, 1\}^n$.

Pour $x, a \in \{0, 1\}$, on note $x^a = x$ si $a = 1$ et $x^a = \neg x$ sinon. Pour $f \in \Gamma_2$, on dira alors que f est de type

- ET s'il existe $a, b, c \in \{0, 1\}$ tels que $f(x_1, x_2) = (x_1^a \wedge x_2^b)^c$ pour tous x_1, x_2 dans $\{0, 1\}$;
- XOR s'il existe $a \in \{0, 1\}$ tel que $f(x_1, x_2) = (x_1 \oplus x_2)^a$ pour tous x_1, x_2 dans $\{0, 1\}$.

Question 2.1.

1. Parmi toutes les fonctions de Γ_2 , quelles sont celles qui sont de type ET? De type XOR? Des deux types simultanément?
2. Montrer qu'une fonction de type XOR est affine. Y a-t-il d'autres fonctions affines dans Γ_2 ?
3. Que vaut $(x_1^a \wedge x_2^b)^c$ pour $x_1 = \neg a$? Que vaut $(x_1 \oplus x_2)^a$ pour $x_1 = x_2$?

Soient f et f' dans Γ_n . On dit que f' est une *spécialisation* de f s'il existe un sous-ensemble $S \subset \{1, \dots, n\}$ et des constantes $C = \{c_s \in \{0, 1\} \mid s \in S\}$ satisfaisant les conditions suivantes : pour tout $(x_1, \dots, x_n) \in \{0, 1\}^n$, $f'(x_1, \dots, x_n) = f(x'_1, \dots, x'_n)$, avec $x'_s = c_s$ si $s \in S$ et $x'_s = x_s$ sinon. On dit que f' *fixe les variables* x_s , pour $s \in S$.

Question 2.2.

1. Montrer par récurrence sur $n \geq 2$ que si $f \in \Gamma_n$ n'est pas affine, il existe une spécialisation f' de f qui fixe toutes les variables sauf deux d'entre elles, x_i et x_j , et telle que $f'(x_1, \dots, x_n) = (x_i^a \wedge x_j^b)^c$, pour des constantes a, b, c dans $\{0, 1\}$.
2. Pour $n \geq 1$, montrer que si $f \in \Gamma_n$ n'est pas monotone, il existe une spécialisation f' de f qui fixe toutes les variables sauf une d'entre elles, x_i , et telle que $f'(x_1, \dots, x_n) = \neg x_i$.

Soit Ω un ensemble fini de fonctions booléennes. Un *circuit* $C = (G, g)$ sur Ω est la donnée

- d'un graphe sans cycle $G = (S, A)$. On suppose que $o(i) = i$ forme un ordre topologique sur G et que les entrées sont numérotées de 1 à n .

- d'une fonction $g : S \rightarrow \Gamma$ telle que $g(s) \in \Omega \cap \Gamma_{\text{in}(s)}$ si $\text{in}(s) > 0$ et, si s est une entrée, $g(s)(x_1, \dots, x_n) = x_s$ pour $(x_1, \dots, x_n) \in \{0, 1\}^n$.

On associe à un circuit $C = (G, g)$ à n entrées une application $h : S \rightarrow \Gamma_n$ définie ainsi :

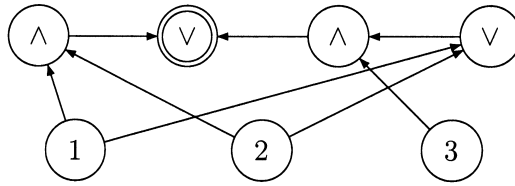
- pour toute entrée s de G , $h(s) = g(s)$;
- si s est un sommet interne, et si $(s_1, s), \dots, (s_k, s)$ sont les arêtes d'extrémité s , avec $s_1 < \dots < s_k$, on définit $h(s)$ par $h(s)(x) = g(s)(h(s_1)(x), \dots, h(s_k)(x))$ pour tout $x \in \{0, 1\}^n$.

Intuitivement, au sommet s , on applique la fonction $g(s)$ aux valeurs calculées dans les sommets précédents. On dit que le sommet s *calcule* la fonction $h(s)$. Une fonction $f \in \Gamma$ est *calculée* par un circuit C sur Ω s'il existe un sommet s du graphe associé tel que $h(s) = f$.

La *taille* d'un circuit est le nombre de sommets internes de G (c'est-à-dire de sommets qui ne sont pas des entrées). La *profondeur* d'un circuit est la longueur du plus long chemin dans G .

Question 2.3.

1. Dans l'exemple suivant, indiquer quelle fonction est calculée par le sommet entouré deux fois (on ne donne que la numérotation des noeuds d'entrée, aucune ambiguïté n'étant possible).



2. Pour x dans $\{0, 1\}^n$, on définit $f_x \in \Gamma_n$ par $f_x(x') = 1$ si et seulement si $x' = x$. Montrer que f_x peut être calculée par un circuit sur $\{0, 1, \vee, \wedge, \neg\}$ de taille $O(n)$ et de profondeur $O(\log(n))$, en utilisant une approche de type "diviser pour régner".
3. En déduire que pour tout $n \geq 1$, toute fonction dans Γ_n peut être calculée par un circuit sur $\{0, 1, \vee, \wedge, \neg\}$. Quelle est la taille de ce circuit ? Sa profondeur ?

Question 2.4. Un ensemble $\Omega \subset \Gamma$ est *complet* si toute fonction de Γ peut être calculée par un circuit sur Ω .

1. Réinterpréter le résultat de la question 2.3.3 en termes d'ensemble complet. Donner un ensemble complet, minimal au sens de l'inclusion.
2. On va caractériser les ensembles Ω complets. Considérons les conditions suivantes :
 - (i) il existe $f \in \Omega$ telle que $f(0, \dots, 0) = 1$ (ici, $0, \dots, 0$ signifie que tous les arguments de f sont mis à 0) ;
 - (ii) il existe $f \in \Omega$ telle que $f(1, \dots, 1) = 0$ (même remarque que ci-dessus) ;
 - (iii) il existe $f \in \Omega$ non monotone ;
 - (iv) il existe $f \in \Omega$ non auto-adjointe ;
 - (v) il existe $f \in \Omega$ non affine ;
 - (vi) il existe un circuit C sur Ω à une entrée et un sommet interne s du graphe associé tels que s calcule la fonction identité $x \mapsto x \in \Gamma_1$.
 - (a) Montrer que ces conditions sont nécessaires.
 - (b) Réciproquement, on suppose ces conditions satisfaites.
 - Montrer en utilisant (i)–(iii) et (vi) qu'on peut calculer la fonction $x \mapsto \neg x \in \Gamma_1$ par un circuit sur Ω .
 - Montrer en utilisant (iv) et (vi) qu'on peut calculer les fonctions **0** et **1** par des circuits sur Ω .
 - Montrer en utilisant (v) qu'on peut calculer par un circuit sur Ω une fonction de type ET. Conclure.

3 Bornes supérieures et bornes inférieures

Dans toute cette partie, on pose $\Omega = \Gamma_1 \cup \Gamma_2$: tous les sommets des graphes considérés sont donc de degré entrant 0, 1 ou 2. La première question traite un exemple de borne supérieure sur le coût du calcul de certaines fonctions ; la fin du problème aborde des questions de bornes inférieures.

Une *matrice booléenne* $X = (x_{i,j})_{1 \leq i,j \leq n}$ de taille n est la donnée de n^2 valeurs $x_{i,j}$ dans $\{0, 1\}$. Si $X = (x_{i,j})_{1 \leq i,j \leq n}$ et $X' = (x'_{i,j})_{1 \leq i,j \leq n}$ sont deux telles matrices, on note $Y = XX'$ la matrice booléenne de taille n définie par $Y = (y_{i,j})_{1 \leq i,j \leq n}$ avec

$$y_{i,j} = (x_{i,1} \wedge x'_{1,j}) \vee (x_{i,2} \wedge x'_{2,j}) \vee \cdots \vee (x_{i,n} \wedge x'_{n,j}).$$

On note par ailleurs $Z = X + X'$ la matrice booléenne de taille n définie par $Z = (z_{i,j})_{1 \leq i,j \leq n}$ avec $z_{i,j} = x_{i,j} \vee x'_{i,j}$. Pour $m \geq 0$, on définit les puissances X^m par récurrence : X^0 est la matrice I_n , qui a toute ses entrées nulles, à l'exception de la diagonale $(x_{i,i})_{1 \leq i \leq n}$, remplie par des 1 ; pour $m \geq 0$, on pose $X^{m+1} = XX^m$.

Question 3.1.

1. Soit $G = (S, A)$ un graphe à n sommets. Sa *matrice d'adjacence* est la matrice booléenne $X = (x_{i,j})_{1 \leq i,j \leq n}$ de taille n telle que $x_{i,j} = 1$ si et seulement si $(i, j) \in A$. Caractériser les éléments de $(I_n + X)^m$ en fonction des chemins dans G .
2. Pour $1 \leq i, j \leq n$, soit $\text{Acc}_{i,j} \in \Gamma_{n^2}$ la fonction qui à la matrice d'adjacence d'un graphe G , de taille n , associe 1 si et seulement si $i = j$ ou s'il existe un chemin de i à j dans G . Montrer que pour tous i, j , $\text{Acc}_{i,j}$ peut se calculer par un circuit sur Ω de taille $O(n^3 \log(n))$ et de profondeur $O(\log(n)^2)$.

Pour une fonction booléenne f , on note $L_\Omega(f)$ la taille minimale d'un circuit qui calcule f sur Ω , si un tel circuit existe (sinon, on pose $L_\Omega(f) = \infty$). Un circuit qui calcule f sur Ω est *optimal* pour f si sa taille vaut $L_\Omega(f)$.

Question 3.2.

1. Soit G un graphe sans cycle, ayant n entrées, m sorties (toutes distinctes des entrées) et p sommets (internes ou non) de degré sortant supérieur ou égal à 2. Montrer que si tout sommet interne de G a un degré entrant inférieur ou égal à 2, alors G contient au moins $n - m + p$ sommets internes.

Indication : on pourra compter le nombre d'arêtes de deux façons différentes.

2. Soit f dans Γ_n et soit $C = (G, g)$ un circuit sur Ω qui calcule f . Montrer les assertions suivantes :
 - (a) Si C est optimal pour f , alors :
 - si s_1 et s_2 sont deux sommets distincts de G , $h(s_1) \neq h(s_2)$;
 - il existe un unique sommet s de G tel que $h(s) = f$ et c'est une sortie de G ;
 - il existe au plus un sommet interne de G qui soit une sortie.
 - (b) On suppose que f n'est ni constante, ni de la forme $(x_1, \dots, x_n) \mapsto \neg x_i$. On suppose également que G a k sommets internes $s_1, \dots, s_k$ pour lesquels $g(s_i)$ est dans Γ_1 ou n'est ni de type ET ni de type XOR. Montrer qu'alors $L_\Omega(f) + k$ est plus petit que la taille de C .

Question 3.3. Soit $k \geq 1$. Pour $\alpha = (\alpha_1, \dots, \alpha_k)$ dans $\{0, 1\}^k$, on note $\bar{\alpha}$ l'entier $1 + \sum_{1 \leq i \leq k} \alpha_i 2^{i-1}$. Posant $n = 2^k$, on définit la fonction $f_k \in \Gamma_{k+n}$ par

$$\begin{aligned} f_k : \quad & \{0, 1\}^{k+n} && \rightarrow \{0, 1\} \\ & (\alpha, x) = (\alpha_1, \dots, \alpha_k, x_1, \dots, x_n) && \mapsto x_{\bar{\alpha}}. \end{aligned}$$

Le but de cette question est de donner une borne inférieure sur $L_\Omega(f_k)$. Pour cela, pour $1 \leq z \leq n$, on définit $A_z \subset \Gamma_{k+n}$ par

$$f \in A_z \iff \exists Z \subset \{1, \dots, n\} \text{ de cardinal } z \text{ tel que } \forall (\alpha, x), \bar{\alpha} \in Z \implies f(\alpha, x) = x_{\bar{\alpha}}.$$

Soit f dans A_z , avec z dans $\{2, \dots, n\}$, soit C un circuit sur Ω qui calcule optimalement f et soit $\alpha \in \{0, 1\}^k$ tel que $i = \bar{\alpha}$ est dans un ensemble Z associé à f selon la définition ci-avant. En fixant x_i , construire une fonction f' dans A_{z-1} telle que $L_\Omega(f') \leq L_\Omega(f) - 2$. En déduire que $L_\Omega(f_k) \geq 2n - 2$.

Indication : on pourra distinguer selon le degré sortant de l'entrée correspondant à x_i . Si ce degré est 1, on discutera selon la nature du fils et on utilisera la question 2.1.3.

Question 3.4. Soit f dans Γ_n , pour $n \geq 3$. On suppose qu'il existe $Z \subset \{1, \dots, n\}$ de cardinal au moins 2, satisfaisant la propriété suivante : pour tous $i, j \in Z$, avec $i \neq j$, il existe des spécialisations $f'_{i,j}$ et $f''_{i,j}$ de f qui fixent toutes les variables sauf x_i et x_j , et des constantes $a_{i,j}, b_{i,j}, c_{i,j}, d_{i,j} \in \{0, 1\}$ telles que

$$f'_{i,j}(x_1, \dots, x_n) = (x_i^{a_{i,j}} \wedge x_j^{b_{i,j}})^{c_{i,j}} \quad \text{et} \quad f''_{i,j}(x_1, \dots, x_n) = (x_i \oplus x_j)^{d_{i,j}}.$$

1. Soit C un circuit optimal qui calcule f sur Ω . Soient i et j dans Z , avec $i \neq j$, et w_i, w_j des chemins de l'entrée i (respectivement l'entrée j) à s_f , l'unique sommet de C qui calcule f . On note s le sommet de plus petit indice commun aux deux ensembles $S(w_i)$ et $S(w_j)$. Montrer qu'il existe, dans $S(w_i)$ ou dans $S(w_j)$, un sommet t d'indice strictement plus petit que celui de s et de degré sortant supérieur ou égal à 2.
2. Montrer qu'il existe $|Z| - 1$ éléments $i_1, \dots, i_{|Z|-1}$ tels que, pour tout ℓ , $1 \leq \ell < |Z|$, tout chemin de i_ℓ à s_f contient un sommet de degré sortant au moins 2. Montrer de plus qu'on peut choisir ces sommets de sorte qu'ils soient tous distincts.
En déduire l'inégalité $L_\Omega(f) \geq 2|Z| - 2$.
3. Pour $c \in \{1, \dots, n - 1\}$, que dire de la fonction f_c définie par $f_c(x_1, \dots, x_n) = 1$ si et seulement si $\sum_{1 \leq i \leq n} x_i = c$?

Question 3.5. Montrer que pour n assez grand, il existe $f_n \in \Gamma_n$ telle que $L_\Omega(f_n) > 2^{n-1}/n$.

Indication : on pourra compter le nombre d'éléments dans Γ_n et utiliser une majoration sur le nombre de circuits de taille L à n entrées qu'on peut construire sur Ω .