Sujets de révisions pas étoile

Logique

Exercice 1 *Syntaxe et sémantique de la logique propositionnelle*

Un voyageur navigue dans un archipel à la recherche de l'île Maya. Les îles de cet archipel sont occupées par d'étranges habitants : les Pires, qui mentent systématiquement ; et les Purs, qui disent toujours la vérité. Sur chaque île, le voyageur rencontre deux habitants, A et B qui acceptent de lui parler.

1. Sur la première île, A et B tiennent les propos suivants :

   A : B est un Pur et nous sommes sur l'île Maya.

   B : A est un Pire et nous sommes sur l'île Maya.

   a) Modéliser la situation à l'aide d'une formule F_1 du calcul propositionnel.

   b) Mettre F_1 sous forme normale conjonctive sans utiliser de table de vérité.

   c) Appliquer l'algorithme de Quine à la FNC obtenue à la question précédente et en déduire si le voyageur est arrivé sur l'île Maya et, si possible, la nature de A et B.

2. Sur la deuxième île, A et B tiennent les propos suivants :

   A : "Nous sommes deux Pires, et nous sommes sur l'île Maya".

   B : L'un de nous au moins est un Pire, et nous ne sommes pas sur l'île Maya".

   a) Modéliser la situation à l'aide d'une formule F_2 du calcul propositionnel.

   b) Etablir la table de vérité de F_2 et en déduire une formule équivalente à F_2 sous forme normale disjonctive.

   c) Déterminer si le voyageur est arrivé sur l'île Maya.

Exercice 2 *Vocabulaire autour de la logique du premier ordre*

Répondre aux questions suivantes en sachant que x, y, z sont des symboles de variables et que l'expression suivante est une formule du premier ordre :

$$F = \forall x \, \exists y \, f(g(x, y), a, z) \wedge \forall z \, f(x, g(x, Q(a)), z)$$

1. Pour chaque symbole dans F, dire si c'est un symbole de fonction ou un symbole de relation et donner son arité.
2. Déterminer l'ensemble des termes et des formules atomiques de F.
3. Pour chaque occurrence de variable dans F, indiquer si elle est libre ou liée ; dans le cas où elle est liée, préciser par quel quantificateur.

Exercice 3 *Déduction naturelle*

1. a) Montrer en déduction naturelle le théorème suivant : $(\neg P \vee Q) \Rightarrow (P \Rightarrow Q)$.

   b) Peut on prouver la réciproque en déduction naturelle ? Justifier.

2. a) Montrer en déduction naturelle que le séquent $\vdash (\forall x \, A \vee \forall x \, B) \Rightarrow \forall x \, (A \vee B)$ est valide.

   b) Qu'en est-il du séquent $\vdash \forall x \, (A \vee B) \Rightarrow (\forall x \, A \vee \forall x \, B)$?

Analyse d'algorithmes

Exercice 4 *Analyses de complexité*

On définit la suite de Fibonacci par $f_0 = f_1 = 1$ et $\forall n \geq 2$, $f_n = f_{n-1} + f_{n-2}$.

1. Donner un algorithme récursif naïf permettant de calculer le n-ème terme de cette suite.
2. On note N_n le nombre d'appels à cet algorithme sur l'entier n. Exhiber une relation de récurrence vérifiée par $(N_n)_{n\in\mathbb{N}}$. En déduire que $\forall n \in \mathbb{N}$, $N_n = 2f_n - 1$.
3. En déduire la complexité de cet algorithme. Qu'en penser ?
4. Donner un algorithme de complexité linéaire en n permettant de calculer f_n.

Dans la suite, on note $F = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$

5. Déterminer (et prouver) une expression de F_n en fonctions des termes de la suite de Fibonacci.
6. En déduire une méthode permettant de calculer f_n en temps $O(\log n)$.

Exercice 5 *Graphe de flot de contrôle*

On considère le programme suivant qui prend en entrée un entier, deux tableaux de taille identique et leur taille.

```
void corresponding_value(int v, int* a1, int* a2, int n){
    int r = -1;
    int i = 0;
    while(i < n)
        if (a1[i] == v){
            r = i;
        }
        i = i + 1;
    }
    return a2[r];
}
```

1. Indiquez ce que fait ce programme.
2. Construisez son graphe de flot de contrôle.
3. Indiquez le chemin obtenu avec l'entrée $\mathsf{v} = 1$, $\mathsf{a1} = \{3,5,1\}$, $\mathsf{a2} = \{5,1,3\}$, $\mathsf{n} = 3$.
4. Commentez la couverture des sommets et des arcs que permet ce chemin.
5. Un jeu de test contenant uniquement ce chemin vous semble-t-il suffisant pour tester le programme ?

Stratégies algorithmiques

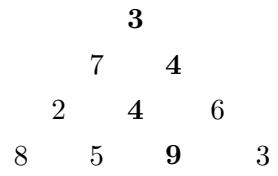
Exercice 6 *Diviser pour régner*

On considère un tableau A de taille n. On pourra dans un premier temps supposer que n est une puissance de deux. Un élément x de A est dit majoritaire si et seulement si A contient strictement plus de $n/2$ occurrences de x.

1. Donner un algorithme naïf permettant de calculer s'il existe un élément majoritaire dans un tableau donné et étudier sa complexité temporelle pire cas et meilleur cas.
2. Ecrire un algorithme occurrences permettant de calculer le nombre d'occurrences de x dans un tableau A entre les indices i et j. Etudier sa complexité.
3. Supposons qu'on divise A en deux parties égales et qu'on soit en mesure de savoir sur chacune des moitiés si elle comprend un élément majoritaire et si oui quelle est sa valeur. Expliquer comment calculer la valeur majoritaire de A si elle existe à partir de cette connaissance.
4. Ecrire un algorithme majoritaire prenant en entrée un tableau A et deux indices dans ce tableau i, j et qui renvoit le couple (Vrai, x) si x est majoritaire dans $A[i, j]$ et (Faux, -1) si $A[i, j]$ ne contient pas d'élément majoritaire en utilisant la question précédente.
5. On note $C(n)$ le nombre d'opérations pire cas effectuées sur l'entrée $(A, 1, n)$ par majoritaire. Déterminer une relation de récurrence sur $C(n)$ et en déduire la complexité de majoritaire.

Exercice 7 *Programmation dynamique*

On considère des entiers naturels disposés sur un triangle équilatéral de hauteur n. La figure suivante donne un exemple pour un tel triangle lorsque $n = 4$:



On appelle chemin une suite de nombres du triangle obtenue en se déplaçant vers les nombres adjacents de la ligne inférieure à partir du sommet et finissant sur un nombre de la base. Un chemin est indiqué en gras sur la figure précédente. On cherche à calculer la valeur maximale d'un chemin reliant le sommet à la base d'un triangle de hauteur n donné. Un triangle sera représenté par une matrice $n \times n$ contenant à la ligne i les coefficients de la i-ème ligne du triangle.

1. Résoudre le problème dans le cas du triangle ci-dessus.
2. Combien y a-t-il de chemins possibles dans le triangle ? En déduire la complexité d'un algorithme naïf (brièvement décrit) résolvant ce problème : est-il envisageable de procéder ainsi ?
3. Proposer un algorithme dynamique permettant de résoudre ce problème. On pourra établir une relation de récurrence sur $f(i, j)$, la valeur maximale d'un chemin passant du sommet à la position (i, j).
4. Etudier la complexité temporelle et spatiale de cet algorithme.
5. Peut on modifier l'algorithme de sorte à calculer un chemin de valeur maximale en plus de ladite valeur sans déteriorer sa complexité ?

Exercice 8 *Algorithmes gloutons et d'approximation*

On considère le problème d'optimisation SUBSET SUM suivant :

$$\begin{cases} \textbf{Entrée} : \text{Un tableau } T = [t_1, ..., t_n] \text{ d'entiers naturels et } C \in \mathbb{N}. \\ \textbf{Solution} : \text{Un sous ensemble } I \subset [\![1, n]\!] \text{ tel que } \sum_{i \in I} t_i \leq C. \\ \textbf{Optimisation} : \text{Maximiser } \sum_{i \in I} t_i. \end{cases}$$

La version décisionnelle de ce problème est NP-complète. On propose d'approcher une solution à SUBSET SUM à l'aide de l'algorithme glouton suivant :

```
S ← 0 et I ← ∅
Pour i allant de 1 à n
    Si S + t_i ≤ C
        S ← S + t_i
        I ← I ∪ {i}
Renvoyer I
```

1. Montrer que cet algorithme n'est pas un algorithme d'approximation à facteur constant pour SUBSET SUM.

On modifie l'algorithme précédent de sorte à considérer les t_i par ordre décroissant de valeur (plutôt que par ordre croissant d'indice) : cela donne naissance à un algorithme qu'on note A.

2. Déterminer la complexité temporelle de A.
3. Montrer que A est une 1/2-approximation de SUBSET SUM.
4. On vient de montrer que le facteur d'approximation de A est au moins égal à 1/2. Est-il meilleur ?

Exercice 9 *Algorithmes probabilistes*

On considère l'opération suivante : étant donnés n éléments distincts et comparables, les insérer dans un arbre binaire de recherche (ABR).

1. Quelle est la complexité de la recherche d'un élément dans l'ABR résultant de cette opération au pire cas en fonction de n ? Et au meilleur cas ? Donner un ordre d'insertion lorsque les éléments sont ceux de $[\![1, 7]\!]$ pour lequel le pire cas est un atteint et un ordre pour lequel le meilleur cas est atteint.

Afin d'éviter le pire cas, on insère les n éléments en les choisissant dans un ordre aléatoire : on considère que les $n!$ permutations possibles des éléments sont des ordres d'insertion équiprobables. On note H_n la variable aléatoire correspondant à la hauteur de l'arbre obtenu, $X_n = 2^{H_n}$, R_n le rang de la racine de l'arbre parmi les n éléments (c'est-à-dire sa position si les n éléments étaient triés) et $Z_{n,i}$ la variable aléatoire valant 1 si $R_n = i$ et 0 sinon.

2. Montrer que, si $R_n = i$ alors $X_n = 2\max(X_{i-1}, X_{n-i})$.
3. Montrer que $X_n = \sum_{i=1}^{n} Z_{n,i} \times 2\max(X_{i-1}, X_{n-i})$.
4. En observant que $Z_{n,i}$ est indépendante de X_{i-1} et X_{n-i}, montrer que $\mathbb{E}[X_n] \leq \frac{4}{n} \sum_{k=0}^{n-1} \mathbb{E}[X_k]$.

On admet qu'une telle relation de récurrence implique que $\mathbb{E}[X_n] \leq \frac{1}{4}\binom{n+3}{n}$.

5. L'inégalité de Jensen garantit que si f est une fonction convexe sur un intervalle I et X est une variable aléatoire sur I alors $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$. Déduire de ce qui précède que $\mathbb{E}[H_n] \leq O(\log n)$ et conclure.

Algorithmique

Exercice 10 *Algorithmes de compression*

1. Compresser la chaîne "charabiabaragouin" à l'aide de l'algorithme de Huffman.
2. Décompresser la chaîne

   1101111100101110111010010110110011100101110100111001101010001101110111100101000

   sachant qu'elle a été compressée à l'aide de l'algorithme de Huffman et que la clé de codage est la suivante :

Lettre	a	d	e	i	n	o	s	t	u
Code	0010	11	0100	011	101	100	000	0011	0101

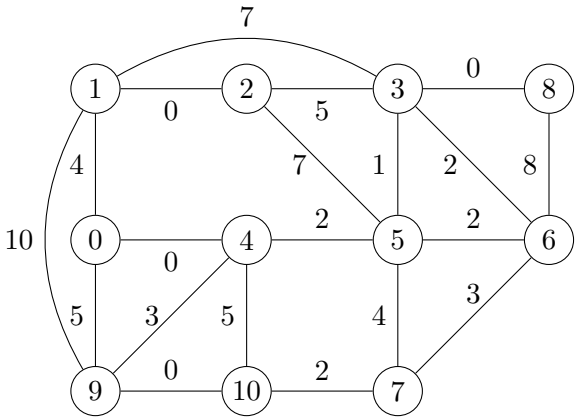
3. Compresser la chaîne AABAACBAAAABBCBAC à l'aide de l'algorithme LZW en prenant comme dictionnaire de base celui qui suit :

Chaine	A	B	C
Code	1	2	3

4. Sachant que le dictionnaire initial est constitué des lettres latines minuscules codées par leur position dans l'alphabet (ainsi, la lettre a est associée au code 1 et la lettre z au code 26), décompresser via l'algorithme LZW la chaîne suivante :

   16 1 16 15 21 19 27 27 19 28 30 24 33 1 19

Exercice 11 *Algorithmique des graphes*



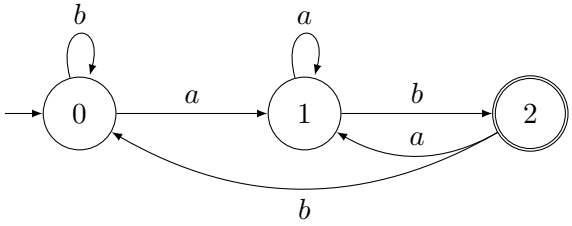
1. Effectuer un parcours en profondeur du graphe depuis 0 en parcourant d'abord les petits sommets.
2. En utilisant l'algorithme de Dijkstra, déterminer un plus court chemin de 1 à 7.
3. En utilisant l'algorithme de Kruskal, déterminer un arbre couvrant minimal pour ce graphe.
4. Orienter les arêtes de poids pair vers le plus petit des deux sommets et les arêtes de poids impair vers le plus grand. Déterminer les composantes fortement connexes du graphe résultant via l'algorithme de Kosaraju et en déduire le graphe de CFC.

Exercice 12 *Algorithmes sur les automates*

1. a) En appliquant l'algorithme de Berry-Sethi, construire un automate reconnaissant le langage L dénoté par $(a+c)^\star abb + (a+c)^\star$.

   b) En déduire un automate reconnaissant L^c.

2. a) En appliquant la méthode d'élimination des états et en supprimant les états par ordre croissant de numéro, donner une expression rationnelle pour le langage reconnu par l'automate suivant :



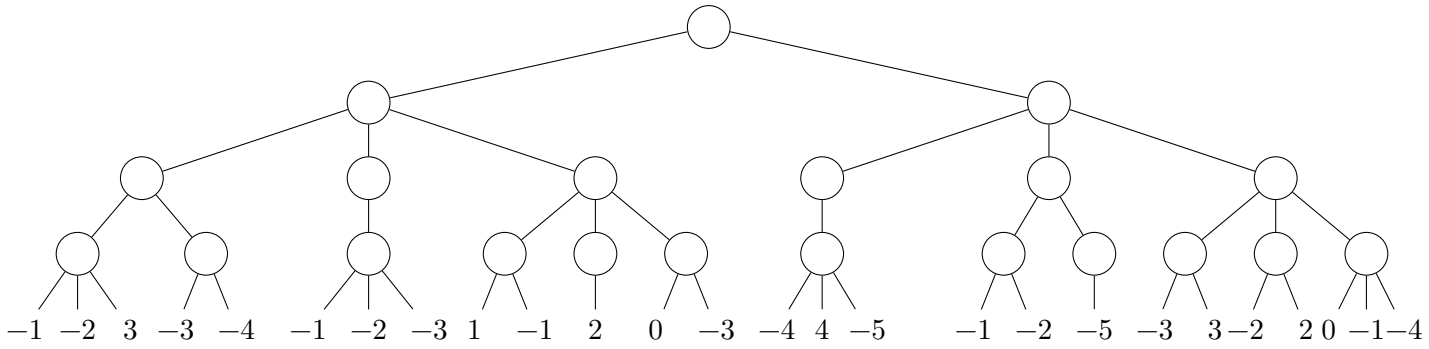
   b) Donner une description simple des mots reconnus par cet automate et en déduire une autre expression rationnelle pour le langage qu'il reconnaît.

Exercice 13 *Min-max et élagage*

Déterminer le score de la racine étant données les valeurs de l'heuristique aux feuilles de l'arbre suivant :

1. En supposant que la racine est un état du joueur qui veut maximiser le score de la racine.
2. En supposant que la racine est un état du joueur qui veut le minimiser.

Identifier les branches qui n'auraient pas été explorées avec un élagage alpha-beta dans le premier cas.



Langages formels

Exercice 14 *Langages quelconques*

Dans ce qui suit, A, B, C sont des langages sur un même alphabet.

1. Déterminer en justifiant quelles sont les affirmations correctes parmi :

   (1) $(A^\star)^\star = A^\star$.
   (2) $A(B + C) = AB + AC$.
   (3) $A(B \cap C) = AB \cap AC$.
   (4) $(A + B)^\star = (A^\star B^\star)^\star$.

On définit à présent la racine carrée d'un langage L sur un alphabet Σ par

$$\sqrt{L} = \{u \in \Sigma^\star \mid u^2 \in L\}$$

1. Déterminer quelles sont les inclusions vraies parmi les suivantes :

   (1) $L \subset \sqrt{L^2}$.
   (2) $\sqrt{L^2} \subset L$.
   (3) $L \subset (\sqrt{L})^2$.
   (4) $(\sqrt{L})^2 \subset L$.
   (5) $(\sqrt{L})^2 \subset \sqrt{L^2}$
   (6) $\sqrt{L^2} \subset (\sqrt{L})^2$.

Exercice 15 *Langages rationnels*

1. Les langages suivants sont-ils rationnels sur l'alphabet $\{a, b\}$? Justifier.

   a) $L_1 = \{a^n b^m \mid n < m\}$
   b) $L_2 = \{a^n b^m \mid n + m \leq 1024\}$
   c) $L_3 = \{a^n b^m \mid n \neq m\}$
   d) $L_4 = \{a^n b^m \mid n \equiv m \bmod 2\}$
   e) $L_5 = \{a^p \mid p \text{ est premier}\}$

Exercice 16 *Langage engendré par une grammaire*

Dans cet exercice, l'alphabet est $\Sigma = \{a, b\}$. On considère le langage L des mots dans le complémentaire de $\{m \in \Sigma^\star \mid \exists u \in \Sigma^\star, m = uu\}$.

1. Donner deux exemples de mots dans L, l'un de longueur 4, l'autre de longueur 5.
2. Montrer que tout mot de longueur impaire est dans L et que tout mot $m_1 ... m_{2n}$ de longueur paire appartient à L si et seulement si il existe un indice $i \in [\![1, n]\!]$ tel que $m_i \neq m_{n+i}$.

On considère la grammaire algébrique G d'axiome S dont les règles sont :

$$\begin{aligned} S &\to A \mid B \mid AB \mid BA \\ A &\to a \mid aAa \mid aAb \mid bAa \mid bAb \\ B &\to b \mid aBa \mid aBb \mid bBa \mid bBb \end{aligned}$$

3. Décrire le langage $L(G, A)$ (resp. $L(G, B)$) des mots qu'il est possible de dériver à partir de A (resp. B).
4. Montrer que tout mot de longueur impaire est dans $L(G)$.
5. Montrer que tout mot $m \in L$ de longueur paire est dans $L(G)$.
6. Montrer que le langage L est algébrique.

Jeux et IA

Exercice 17 *Jeux et couplages*

Le jeu de Slither est un jeu à deux joueurs qui se joue sur un graphe connexe. Tour à tour, chaque joueur doit choisir un sommet de G qui n'a jamais été choisi par aucun des joueurs et qui est voisin de celui choisi précédemment (autrement, dit, la suite de sommets choisis doit former un chemin élémentaire dans G). Le premier joueur qui ne peut plus jouer a perdu.

Montrer que si G admet un couplage parfait alors le second joueur a une stratégie gagnante.

Exercice 18 *Modélisation d'un jeu à deux joueurs*

On considère un jeu dont le support est une rangée de n bâtonnets. Tour à tour, chaque joueur enlève de la rangée 1, 2 ou 3 bâtonnets. Le perdant est celui qui prend le dernier bâtonnet.

1. Dessiner le graphe des configurations de ce jeu lorsque $n = 7$.
2. Donner une stratégie gagnante pour le premier joueur dans ce cas en justifiant.
3. Montrer que les configurations où le nombre n de bâtonnets est congru à 1 modulo 4 sont perdantes. Si $n = 20$, vaut-il mieux jouer en premier ou en deuxième ? Et si $n = 21$?

Exercice 19 *ID3*

On considère le problème de classification suivant : des objets sont répartis en deux classes - rond et croix - selon deux attributs prenant des valeurs continues dans $[0, 1]^2$. On pourra représenter ces objets par des points (ronds ou croix) dans le plan.

1. Peut on appliquer l'algorithme ID3 à de telles données afin d'établir un modèle permettant de les classer ? Expliquer votre réponse.
2. Proposer une méthode naïve permettant d'appliquer ID3 à un tel jeu de données. Est-elle raisonnable ?
3. Dans la suite, on souhaite classer les données à l'aide d'un arbre de décision obligatoirement binaire. Donner une méthode moins naïve que celle de la question 2 permettant d'atteindre ce but.
4. Quel résultat obtiendrait-on avec cette méthode sur la figure 1 ? Et sur la figure 2 ? Est-ce satisfaisant ?

FIGURE 1

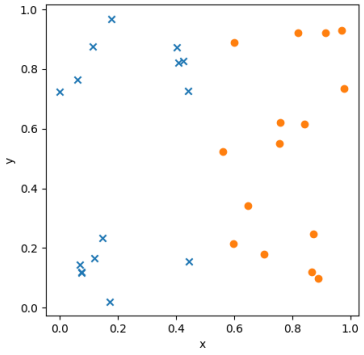
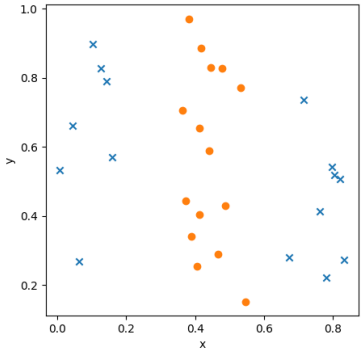


FIGURE 2



5. Proposer une méthode permettant de mieux classer les données dans le cas où celles-ci sont réparties de manière similaire à celles de la figure 2. Argumenter quant à sa terminaison.

6. Dessiner la frontière entre les deux classes que la méthode de la question 5 calculerait sur la figure 3. Est-ce satisfaisant ?
7. Expliquer comment transformer le jeu de données de la figure 3 de sorte à pouvoir tracer une frontière bien plus simple à l'aide de la méthode décrite à la question 5.
8. Et dans le cas de la figure 4, que peut-on faire ?

FIGURE 3

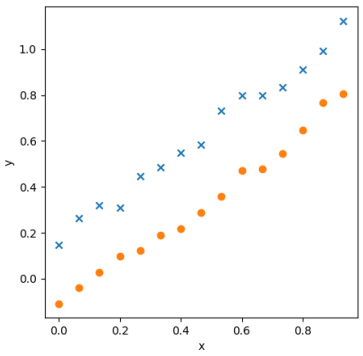
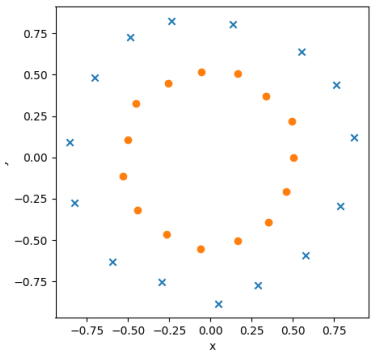


FIGURE 4



Autres

Exercice 20 *Concurrence*

1. On considère les fonctions suivantes ; x et y étant deux variables globales initialement nulles :

$f_1() =$
Si $x = 0$
$x \leftarrow x + 1$
$y \leftarrow y + 1$

$f_2() =$
Si $x = 0$
$x \leftarrow x + 1$
$y \leftarrow y + 2$

   L'instruction $t \leftarrow t + a$ où $a \in \mathbb{N}$ consiste à lire la valeur de t et la stocker dans une variable v locale au fil, ajouter a à v et écraser t avec cette valeur. On suppose que f_1 et f_2 sont exécutées de manière concurrente par deux fils. En fin d'exécuction, est-il possible : que $x = 0$? Que $x = 2$? Que $(x, y) = (1, 3)$?

2. Le code suivant prétend implémenter un mutex :

   ```
   int locked = false;

   void lock()
   {
       while (locked) { /* wait */}
       locked = true;
   }

   void unlock()
   {
       locked = false;
   }
   ```

   Garantit-il l'exclusion mutuelle ?

3. Si on dispose d'une fonction bool test_and_set(bool* ptr) qui permet atomiquement de lire la valeur à l'adresse ptr, d'y écrire la valeur true si *ptr vaut false et de renvoyer l'ancienne valeur de *ptr, peut-on faire mieux ?

Exercice 21 *Problèmes NP-complets*

On considère le problème de décision STABLE suivant :

Entrée : Un graphe $G = (S, A)$ non orienté et $k \in \mathbb{N}$.
Question : Existe-t-il $S' \subset S$ tel que $|S'| = k$ et deux sommets de S' ne sont jamais adjacents dans G ?

Un tel sous ensemble S' s'appelle un stable de taille k.

1. Montrer que STABLE $\in$ NP.

On cherche à présent à montrer que 3SAT $\leq$ STABLE. Soit donc $\varphi = \bigwedge_{i=1}^{m} C_i$ une formule du calcul propositionnel dont les clauses C_i contiennent au plus 3 littéraux et impliquant un ensemble $V = \{v_1, ..., v_n\}$ de variables propositionnelles. On définit le graphe non orienté $G_\varphi = (S, A)$ comme suit :

- Pour chaque occurrence de littéral dans φ, on construit un sommet de S.

- Si $s, t \in S$, on ajoute l'arête $\{s, t\}$ à A si et seulement si (s et t sont des sommets représentant deux littéraux d'une même clause) ou (s et t sont des sommets étiquetés par deux littéraux qui sont la négation l'un de l'autre).

2. Tracer le graphe G_φ associé à $\varphi = (\overline{x} \vee y \vee \overline{z}) \wedge (x \vee \overline{y} \vee z) \wedge (x \vee y \vee z) \wedge (\overline{x} \vee \overline{y})$.
3. Montrer que si G_φ possède un stable de taille m, alors φ est satisfiable par une valuation qu'on explicitera. La réciproque est-elle vraie ?
4. Montrer que STABLE est NP-complet.
5. En déduire que CLIQUE est NP-complet avec CLIQUE le problème défini par :

$$\begin{cases} \textbf{Entrée} : \text{Un graphe } G = (S, A) \text{ non orienté et } k \in \mathbb{N}. \\ \textbf{Question} : \text{Existe-t-il un sous-graphe complet de } G \text{ ayant } k \text{ sommets ?} \end{cases}$$

Exercice 22 *Représentation de flottants*

1. On représente des réels sur 8 bits avec 3 bits d'exposant et 4 de mantisse. Le flottant 01101101 est-il normalisé ? Que vaut-il ? Même question pour le flottant 00001000.
2. Donner la valeur réelle des expressions suivantes et indiquer si un ordinateur fournirait les mêmes réponses : $\sum_{n=1}^{+\infty} \frac{1}{n}$ et $(10^{50} + 1) - 10^{50}$. Expliquer.

Exercice 23 *Graphes*

Si G est un graphe non orienté, on appelle chaîne hamiltonienne dans G tout chemin dans G passant une et une seule fois par chaque sommet et cycle hamiltonien toute chaîne hamiltonienne dont l'origine et l'arrivée sont reliées dans G. Un graphe possédant un cycle hamiltonien est dit hamiltonien.

1. Le graphe complet à n sommets est-il hamiltonien ?

On considère un graphe simple non orienté $G = (S, A)$ ayant $n \geq 3$ sommets et tel que pour toute paire de sommets $\{u, v\}$ non adjacents dans G, $\deg(u) + \deg(v) \geq n$. Supposons que G est non hamiltonien.

2. Montrer qu'il existe un graphe $H = (S', A')$ tel que $S = S'$, $A \subset A'$, H est non hamiltonien mais rajouter une arête à H le rend hamiltonien.
3. Montrer que ce graphe H possède une chaîne hamiltonienne, qu'on note $(s_1, ..., s_n)$.
4. Montrer qu'il existe $1 < i < n$ tel que $(s_1, s_i) \in A$ et $(s_{i-1}, s_n) \in A$. On pourra considérer les ensembles $I = \{i \in [\![2, n]\!] \,|\, (s_1, s_i) \in A\}$ et $J = \{i \in [\![2, n]\!] \,|\, (s_{i-1}, s_n) \in A\}$.
5. Que déduit-on des questions précédentes quant au caractère hamiltonien ou non de G ? *Indication : faire un dessin !*

Exercice 24 *Bases de données*

On considère la base de données d'un établissement scolaire. On y trouve des élèves, chacun appartenant à une classe et des enseignants pouvant intervenir dans une ou plusieurs classes. Les entités du modèle relationnel de cette base et leurs attributs sont :

- `eleves` d'attributs `numero_secu` (le numéro de sécurité sociale de l'élève), `nom`, `prenom` et `classe`.
- `classes` d'attributs `id`, `effectif` et `prof_principal`.
- `profs` d'attributs `numen` (le NUMEN étant un identifiant unique attribué à chaque personnel de l'Education Nationale), `matiere`, `nom` et `prenom`.

1. Construire le diagramme entité-association de cette base.
2. Donner un schéma relationnel de cette base en précisant les clés primaires et étrangères.
3. Donner suivants une requête SQL permettant de calculer...
   a) ... le numéro des classes à strictement plus de 35 élèves.
   b) ... l'identité des 3 premiers enseignants qui enseignent soit le russe soit le latin dans l'ordre décroissant des NUMEN.
   c) ... le nom de toutes les personnes présentes dans l'établissement.
   d) ... le nom du professeur principal de Juliette Béranger.
   e) ... le nombre total d'élèves dans l'établissement.
   f) ... le nombre de professeurs qui sont professeurs principaux (sachant qu'un professeur peut être professeur principal de plusieurs classes à la fois).
   g) ... pour chaque enseignant, son NUMEN et le nombre de classes dans lesquelles il enseigne.
   h) ... le numéro des classes qui ont l'effectif le plus important.