Complexité
Partiel - 2 heures

Les questions indiquées par une étoile ($\star$) signalent des questions sans doute plus difficiles que les autres. Tout document et autre polycopié est interdit.

Le sujet comporte 4 pages et est composé de 4 parties dans l'ensemble indépendantes. Il est ainsi possible de traiter chacune des parties séparemment. Cependant, la dernière question de la troisième partie utilise des définitions données dans la deuxième.

Il est possible d'admettre les résultats d'une question pour résoudre la suivante, il faut pour cela l'indiquer clairement dans la copie. Il n'est pas nécessaire de résoudre toutes les questions pour avoir la note maximale. La rédaction sera prise en compte dans la notation.

1 Classes de complexité

Q 1) Rappeler la définition de la classe de complexité **LOGSPACE**. Montrer que le langage suivantest dans **LOGSPACE** :

$$L_1 = \{\omega \in \{a, b\}^\star | \omega \text{ a strictement plus de } a \text{ que de } b\}$$

Réponse 1) **LOGSPACE** est l'ensemble des langages acceptés par une machine de Turing fonctionnant en espace logarithmique : **LOGSPACE** = **DSPACE**[() $\log n$)

Pour montrer que $L_1 \in$ **LOGSPACE**, il suffit de remarquer que le programme suivant accepte L_1 et fonctionne en espace logarithmique :

```
int f(char t[n])
{
  int a = 0;
  int b = 0;
  int k = 0;
  for (i = 1; i <= n; i++)
  {
    if t[i] == 'a'       a = a +1;
else if t[i] == 'b'  b = b + 1;
  }
 if a > b return 1; else return 0;
}
```

Ce programme utilise 3 variables, chacune variant de 0 à n au maximum, donc chacun de taille $\log n$. Le programme fonctionne donc bien en espace logarithmique et il est clair qu'il accepte L_1.

Q 2) Rappeler la définition de **NP**. Montrer que le langage L_2 suivant est dans **NP**.

$$L_2 = \left\{ a_1 \dots a_n \in \mathbb{N} \middle| \exists I \subseteq [1 \dots n], \sum_{i \in I} a_i = \sum_{i \notin I} a_i \right\}$$

(Autrement dit $a_1 \dots a_n$ est dans L_2 si on peut partager les éléments a_i en deux parties de même somme.)

Réponse 2) **NP** est l'ensemble des langages acceptés par une machine de Turing non déterministe fonctionnant en temps polynomial. $\mathbf{NP} = \cup_i \mathbf{NTIME}[(]n^i)$

Pour montrer que L_2 est dans **NP**, il suffit de considérer l'algorithme non déterministe suivant : sur l'entrée $a_i \dots a_n$, deviner I, puis tester si $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$. Cet algorithme accepte bien L_2. Deviner I prend un temps polynomial puisque I est en espace polynomial en n. Faire le test est également en temps polynomial, donc l'algorithme est bien en temps polynomial.

Q 3) On suppose que $L_2 \in \mathbf{P}$. Donner un algorithme polynomial qui décide, étant donné (a_i) et deux éléments privilégiés a_j et a_k, si on peut partager les éléments a_i en deux ensembles de même somme, j et k se retrouvant dans le même ensemble. (Indice : fusionner)

Réponse 3) Sur l'entrée (a_i), j et k, on considère la suite (b_i) qui est comme la suite (a_i), mais auquelle on a enlevé a_j et a_k et ajouté un nouvel élément b_0 de valeur $b_0 = a_j + a_k$. Alors on voit que $(b_i) \in L_2$ si et seulement si on peut partager les éléments a_i en deux ensembles de même somme, j et k se retrouvant dans le même ensemble. Comme on a un algorithme polynomial pour décider L_2 (c'est l'hypothèse $L_2 \in P$), on a donc un algorithme polynomial qui répond à la question.

$\star$**Q 4)**$\star$ En déduire que si $L_2 \in \mathbf{P}$, il existe un algorithme polynomial qui étant donné (a_i) trouve la répartition en deux ensembles, c'est à dire trouve I tel que $\sum_{i \in I} a_i = \sum_{i \notin I} a_i$

Réponse 4) Question non corrigée.

2 Classe DP

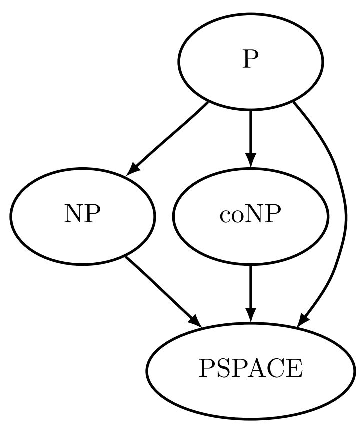
Dans toute cette partie, on suppose que l'alphabet est $\{a, b\}$.

Q 5) Expliquer pourquoi $\mathbf{P} \subseteq \mathbf{NP}$.

Réponse 5) **P** est l'ensemble des langages acceptés par une machine de Turing déterministe fonctionnant en temps polynomial. **NP** est l'ensemble des langages acceptés par une machine de Turing non déterministe fonctionnant en temps polynomial. Une machine déterministe est un cas particulier d'une machine non déterministe : c'est juste une machine non déterministe qui ne fait jamais de choix. De sorte que $\mathbf{P} \subseteq \mathbf{NP}$.

On décide de représenter graphiquement les inclusions connues entre classes par des arêtes dans un graphe : On représente les classes de complexité par un sommet, et on relie le sommet C au sommet D si $C \subseteq D$.

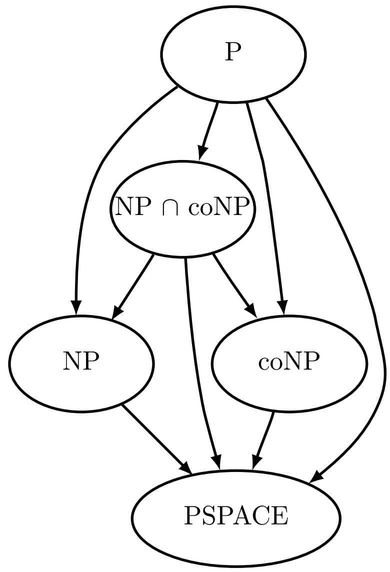
Ainsi, le diagramme suivant :



signifie que $\mathbf{P} \subseteq \mathbf{NP}, \mathbf{P} \subseteq \mathbf{coNP}, \mathbf{P} \subseteq \mathbf{PSPACE}, \mathbf{NP} \subseteq \mathbf{PSPACE}, \mathbf{coNP} \subseteq \mathbf{PSPACE}$.

Q 6) Insérer $\mathbf{NP} \cap \mathbf{coNP}$ dans ce diagramme. (On ne demande pas de justifier les différentes flèches introduites.)

Réponse 6) $\mathbf{P} \subset \mathbf{NP} \cap \mathbf{coNP}$ et $\mathbf{NP} \cap \mathbf{coNP} \subset \mathbf{NP}$ (resp. $\mathbf{coNP}$) ce qui donne le diagramme page suivante.



On définit **DP** de la façon suivante : Un langage L est dans **DP** s'il existe $L_1 \in \mathbf{NP}$ et $L_2 \in \mathbf{coNP}$ tel que $L = L_1 \cap L_2$.

Q 7) Quelle est la différence entre **DP** et $\mathbf{NP} \cap \mathbf{coNP}$?

Réponse 7) $\mathbf{NP} \cap \mathbf{coNP}$ consiste à faire l'intersection des classes, **DP** fait l'intersection des langages contenus dans les classes. En conséquence, $\mathbf{NP} \cap \mathbf{coNP}$ est inclus à la fois dans **NP** et **coNP** alors que ce n'est pas nécessairement le cas de **DP**.

Q 8) En constatant que tout langage L s'écrit $L = L \cap L_\sharp$ où $L_\sharp$ est le langage $L_\sharp = \{a, b\}^\star$, montrer que $\mathbf{NP} \subseteq \mathbf{DP}$ et $\mathbf{coNP} \subseteq \mathbf{DP}$.

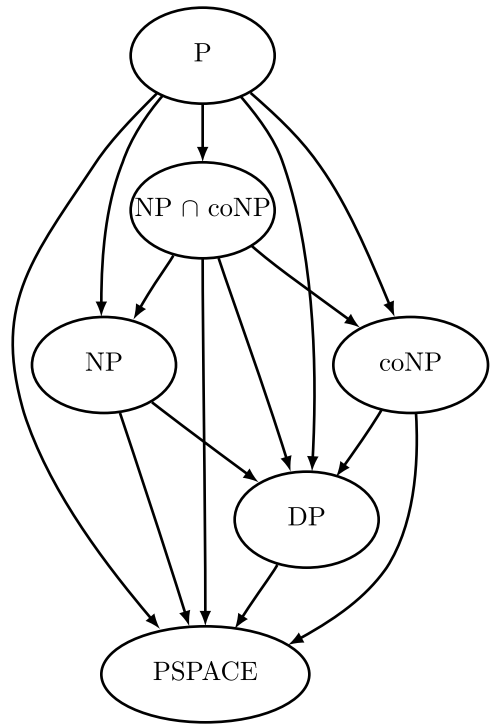
Réponse 8) Soit $L \in \mathbf{NP}$. Alors $L = L \cap L_\sharp$, où $L \in \mathbf{NP}$ et $L_\sharp \in \mathbf{P} \subset \mathbf{coNP}$, de sorte que $L \in \mathbf{DP}$. Ainsi $\mathbf{NP} \subset \mathbf{DP}$. On montre $\mathbf{coNP} \subseteq \mathbf{DP}$ de manière similaire.

Q 9) Montrer que $\mathbf{DP} \subseteq \mathbf{PSPACE}$.

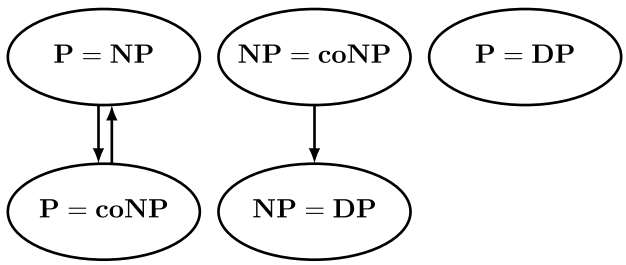
Réponse 9) Soit $L \in \mathbf{DP}$. IL existe $L_1 \in \mathbf{NP}$, et $L_2 \in \mathbf{coNP}$ tel que $L = L_1 \cap L_2$. Comme $L_1 \in \mathbf{NP} \subset \mathbf{PSPACE}$ et $L_2 \in \mathbf{coNP} \subset \mathbf{PSPACE}$, L est l'intersection de deux langages dans **PSPACE**, donc dans **PSPACE**. Ainsi $L \in \mathbf{PSPACE}$, de sorte que $\mathbf{DP} \subset \mathbf{PSPACE}$.

Q 10) Ajouter **DP** dans le diagramme. Justifier brièvement toutes les flèches introduites.

Réponse 10) Toutes les flèches ont été prouvées dans les questions précédentes. Les autres sont des conséquences de celles déjà présentes.



On considère maintenant le diagramme suivant, où les sommets sont des égalités entre classes, et on relie deux sommets par une flèche si une des égalités implique l'autre. Ainsi, le diagramme suivant :



Signifie que $\mathbf{P} = \mathbf{NP} \iff \mathbf{P} = \mathbf{coNP}$ et que $\mathbf{NP} = \mathbf{coNP} \implies \mathbf{NP} = \mathbf{DP}$

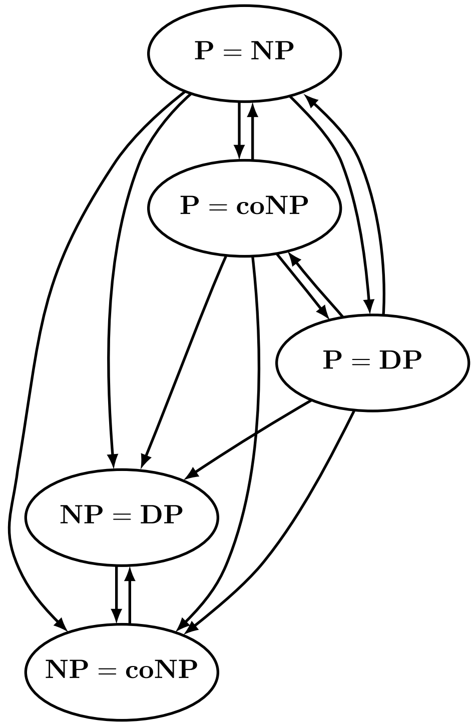
Q 11) Justifier ces trois flèches.

Réponse 11) Si $\mathbf{P} = \mathbf{NP}$, alors $\mathbf{coNP} = \mathbf{coP} = \mathbf{P}$. La réciproque est identique. Si $\mathbf{NP} = \mathbf{coNP}$, alors si $L \in \mathbf{DP}$, $L = L_1 \cap L_2$ avec $L_1 \in \mathbf{NP}$ et $L_2 \in \mathbf{coNP} \subset \mathbf{NP}$, donc L est l'intersection de deux langages **NP** donc dans **NP** de sorte que $L \in \mathbf{NP}$, c'est à dire $\mathbf{DP} \subset \mathbf{NP}$ et comme $\mathbf{NP} \subset \mathbf{DP}$, on a l'égalité.

Q 12) Compléter le diagramme en ajoutant les flèches manquantes. On justifiera chacune des flèches ajoutées (Indiquer au moins 11 nouvelles flèches).

Réponse 12)
- Si $\mathbf{P} = \mathbf{NP}$, alors $\mathbf{NP} = \mathbf{P} = \mathbf{coNP}$ de sorte que $\mathbf{P} = \mathbf{NP} = \mathbf{DP}$, autrement dit $\mathbf{P} = \mathbf{NP}$ implique toutes les autres égalités.
- Si $\mathbf{P} = \mathbf{coNP}$, alors $\mathbf{P} = \mathbf{NP}$, de sorte que $\mathbf{P} = \mathbf{coNP}$ implique toutes les autres égalités d'après l'affirmation précédente.
- Si $\mathbf{P} = \mathbf{DP}$, alors $\mathbf{NP} \subset \mathbf{DP} \subset \mathbf{P}$, donc $\mathbf{P} = \mathbf{NP}$, et on a encore toutes les applications.
- Si $\mathbf{NP} = \mathbf{DP}$, alors $\mathbf{coNP} \subset \mathbf{DP} \subset \mathbf{NP}$, donc $\mathbf{coNP} \subset \mathbf{NP}$, donc $\mathbf{NP} \subset \mathbf{coNP}$ et on a l'égalité, ce qui ajoute une nouvelle flèche

On ajoute tout ça sur le diagramme (désolé pour la disposition)

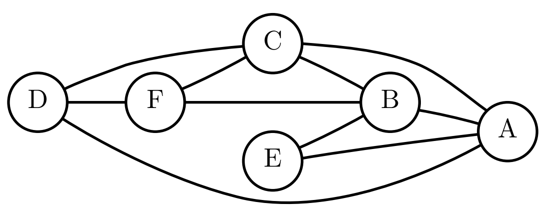


3 Coloriage

On rappelle qu'un graphe est k-coloriable si on peut colorier ses sommets en utilisant au maximum k couleurs, de sorte que deux sommets reliés par une arête aient des couleurs différentes.

Pour tout graphe G, on note $\rho(G)$ le plus petit k tel que G est k-coloriable. De sorte que $\rho(G) \leq k$ si et seulement si G est k-coloriable.

Q 13) Montrer que le graphe suivant est 3-coloriable mais pas 2-coloriable (autrement dit $\rho(G) = 3$).



Réponse 13) Question non corrigée.

Q 14) Montrer que le langage suivant est dans **NP**

$$L = \{(G, k) | \rho(G) \leq k\}$$

Réponse 14) L est l'ensemble des couples (G, k) tel que G est k-coloriable. Pour tester si G est k-coloriable, il suffit de deviner une coloration en k couleurs et vérifier que deux sommets reliés par une arête ont des couleurs différentes. Ceci se fait bien en temps polynomial, de sorte que $L \in$ **NP**.

Q 15) Montrer que le langage suivant est dans **DP**.

$$L = \{(G, k) | \rho(G) = k\}$$

Réponse 15) Soit L_1 le langage suivant $\{(G, k) | \rho(G) \leq k\}$ et L_2 le langage suivant $\{(G, k) | \rho(G) > k - 1\}$ Alors $L = L_1 \cap L_2$. L_1 est dans **NP** et L_2 est dans **coNP** puisque son complémentaire est $\{(G, k) | \rho(G) \leq k - 1\}$ qui est dans **NP** en refaisant la preuve de la question précédente. De sorte que $L \in$ **DP**.

4 Etoile

Si L est un langage sur l'alphabet $\{a, b\}$, on note $L^\star$ l'ensemble des mots qui s'écrivent comme la concaténation de plusieurs mots de L.

Par exemple, si L est l'ensemble des palindromes de longueur paire, le mot $\omega = abbaaabaab \in L^\star$ puisqu'il s'écrit $abba \cdot aa \cdot baab$ donc comme la concaténation de palindromes de longueur paire.

Q 16) Montrer que si $L \in \mathbf{P}$, alors $L^\star \in \mathbf{NP}$.

On cherche maintenant à prouver que si $L \in \mathbf{P}$ alors $L^\star \in \mathbf{P}$. Etant donné un langage L et un mot ω de longueur n, on considère le graphe à n sommets, numérotés de 1 à n, où on relie i à j si et seulement si les lettres i à j de ω forment un mot de L.

Par exemple, si L est l'ensemble des palindromes pairs, et si w est le mot *aabba*, on relie 2 à 5 puisque les lettres 2 à 5 de *aabba* forment le mot *abba* qui est un palindrome de longueur paire.

Q 17) Construire le graphe lorsque L est l'ensemble des palindromes pairs, et pour $\omega = abbaabaaba$.

Q 18) Justifier que si $L \in \mathbf{P}$, on peut construire le graphe en temps polynomial.

$\star$**Q 19)**$\star$ Comment se traduit sur le graphe le fait que $\omega \in L^\star$?

$\star$**Q 20)**$\star$ En déduire que si $L \in \mathbf{P}$, alors $L^\star \in \mathbf{P}$.