Second Concours à l'ENS Paris-Saclay et l'ENS Rennes
Interrogation orale d'informatique
Déroulement de l'épreuve

Le sujet se compose de deux problèmes indépendants. Le jury recommande de traiter les deux sujets, au moins partiellement.

Le temps de préparation est de 2 heures. Ensuite le candidat exposera les résultats obtenus pendant une heure à l'oral. Bien entendu, le jury posera des questions et pourra revenir sur des questions omises par le candidat.

Il est autorisé d'admettre le résultat d'une question pour ne pas rester bloqué lors de la phase de préparation.

Il est enfin demandé au candidat de mentionner en début d'oral quelles questions il a traitées dans les deux problèmes. Cela permettra au jury de gérer au mieux le temps et de permettre au candidat de présenter l'intégralité des résultats obtenus lors de la préparation.

La rigueur du raisonnement scientifique est un point capital dans l'évaluation de l'épreuve.

Automates probabilistes

Soit Σ un alphabet fini et non vide. Un **automate probabiliste** $\mathcal{A}$ sur l'alphabet Σ est la donnée d'un tuple $\mathcal{A} = \langle Q, \{\mathbb{P}_a\}_{a\in\Sigma}, \pi_0, T\rangle$ où

— Q est un ensemble fini d'états ;

— pour tout $a \in \Sigma$, $\mathbb{P}_a$ est la matrice $Q \times Q$ de probabilité des transitions, c'est-à-dire que pour tout $p \in Q$, $\sum_{q\in Q}(\mathbb{P}_a)_{p,q} = 1$;

— $\pi_0 \colon Q \to [0,1]$ est une distribution de probabilité initiale, c'est-à-dire $\sum_{q\in Q}\pi_0(q) = 1$;

— $T \subseteq Q$ est l'ensemble des états terminaux (qu'on représentera avec un double cercle dans les dessins).

On représente le fait que $P_a(p,q) = x$ à l'aide de la flèche $p \xrightarrow{a,x} q$. Étant donné un mot $w = a_0a_1\cdots a_{n-1} \in \Sigma^*$, la **probabilité d'acceptation** du mot w dans l'automate probabiliste $\mathcal{A}$ est définie par

$$\mathbb{P}_{\mathcal{A}}(w) = \sum_{p\in Q}(\pi_0)_p\sum_{q\in T}(\mathbb{P}_w)_{p,q} \qquad \text{avec} \qquad \mathbb{P}_w = \prod_{i=0}^{n-1}\mathbb{P}_{a_i}$$

En particulier, $\mathbb{P}_\varepsilon$ est la matrice identité.

On peut aussi retrouver la valeur $\mathbb{P}_{\mathcal{A}}(w)$ à l'aide de la notion traditionnelle de calcul acceptant : un calcul sur le mot $w = a_0a_1\cdots a_{n-1}$ est une suite de transitions

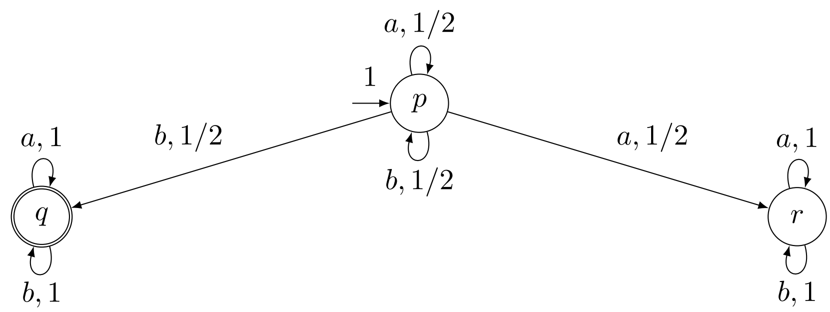
$$p_0 \xrightarrow{a_0,x_0} p_1 \xrightarrow{a_1,x_1} p_2 \cdots p_{n-1} \xrightarrow{a_{n-1},x_{n-1}} p_n$$

Il est dit **acceptant** si $p_n \in T$. Sa probabilité est $(\pi_0)_{p_0}\cdot x_0\cdot x_1\cdots x_{n-1}$. La valeur $\mathbb{P}_{\mathcal{A}}(w)$ peut alors s'obtenir en sommant la probabilité des calculs acceptants sur le mot w.

1. Pour tout mot $w = w_0w_1\dots w_{n-1} \in \{a,b\}^*$, associons une valeur réelle

$$\mathsf{val}(w) = \sum_{i=0}^{n-1}\frac{\mathsf{val}(w_i)}{2^{i+1}}$$

où on note $\mathsf{val}(a) = 0$ et $\mathsf{val}(b) = 1$. En particulier, $\mathsf{val}(\varepsilon) = 0$. Montrer que la probabilité d'acceptation de tout mot $w \in \{a,b\}^*$ dans l'automate $\mathcal{A}$ ci-après (où la distribution initiale est concentrée dans l'état p) est $\mathsf{val}(w)$.



2. Montrer que pour tout automate probabiliste $\mathcal{A}$ et mot $w \in \Sigma^*$, $\mathbb{P}_{\mathcal{A}}(w) \in [0,1]$. La fonction $\mathbb{P}_{\mathcal{A}} \colon \Sigma^* \to [0,1]$ est-elle une distribution de probabilités sur l'ensemble des mots ?

On peut associer à un automate probabiliste des langages, définis par un *seuil* $\theta \in [0,1]$ d'acceptation :

$$\mathcal{L}_{\geq\theta}(\mathcal{A}) = \{w \in \Sigma^* \mid \mathbb{P}_{\mathcal{A}}(w) \geq \theta\}$$

Ces langages sont appelés *langages probabilistes*. Si de plus θ, ainsi que toutes les probabilités apparaissant dans les matrices de transition de $\mathcal{A}$ sont rationnels, alors on parle de *langages probabilistes rationnels* : un tel automate $\mathcal{A}$ est alors appelé *automate probabiliste rationnel*.
3. Décrire le langage $\mathcal{L}_{\geq 1/2}(\mathcal{A})$ pour l'automate $\mathcal{A}$ de la question 1.
4. Montrer que tout langage régulier (reconnu par un automate fini) est un langage probabiliste rationnel.
5. Soit $\mathcal{A}$ un automate probabiliste rationnel. Montrer qu'il existe un automate probabiliste rationnel $\mathcal{A}'$ tel que $\mathcal{L}_{=1/2}(\mathcal{A}) = \mathcal{L}_{\geq 1/4}(\mathcal{A}')$.
6. Montrer que le langage $\{a^n b^n \mid n > 0\}$ est un langage probabiliste rationnel. *On pourra commencer par construire un automate probabiliste qui associe à un mot w de la forme $a^m b^n$ la probabilité $\frac{1}{2} + \frac{1}{2^n} - \frac{1}{2^m}$.*
7. En déduire que les langages rationnels sont un sous-ensemble strict des langages probabilistes rationnels.

Nous allons montrer que le problème de décider le vide des langages probabilistes rationnels est indécidable. Pour cela, on se base sur le **problème de correspondance de Post** (PCP) : étant donné un alphabet Σ et deux morphismes φ_1 et φ_2 de Σ dans $\{a,b\}^+$, existe-t-il un mot $w \in \Sigma^+$ tel que $\varphi_1(w) = \varphi_2(w)$?
8. Montrer qu'on peut restreindre, sans perdre l'indécidabilité, le problème de PCP à des morphismes φ en entrée tels que pour toute lettre $x \in \Sigma$, $\varphi(x) \in (ba + bb)^+$.

On fixe désormais une instance de PCP vérifiant la propriété de la question précédente. On note $\widetilde{u}$ le **miroir** d'un mot u.
9. Montrer que pour tout mot $w \in \Sigma^*$ et $x \in \Sigma$,

$$\mathsf{val}\big(\widetilde{\varphi_1(wx)}\big) = \Big(\mathsf{val}\big(\widetilde{\varphi_1(x)}\big) + \frac{1}{2^{|\varphi_1(x)|}}\Big)\mathsf{val}\big(\widetilde{\varphi_1(w)}\big) + \mathsf{val}\big(\widetilde{\varphi_1(x)}\big)\Big(1 - \mathsf{val}\big(\widetilde{\varphi_1(w)}\big)\Big)$$

10. Utiliser la formule précédente pour construire un automate probabiliste rationnel $\mathcal{A}$ tel que pour tout mot $w \in \Sigma^+$,

$$\mathbb{P}_{\mathcal{A}}(w) = \frac{\mathsf{val}\big(\widetilde{\varphi_1(w)}\big) + 1 - \mathsf{val}\big(\widetilde{\varphi_2(w)}\big)}{2}$$

11. Avec les notations de la question précédente, en déduire que $\mathcal{L}_{=\frac{1}{2}}(\mathcal{A}) = \{\varepsilon\}$ si et seulement si l'instance de PCP n'a pas de solution.
12. À partir de l'automate $\mathcal{A}$ précédent, proposer la construction d'un automate probabiliste rationnel $\mathcal{A}'$ tel que $\mathcal{L}_{=\frac{1}{2}}(\mathcal{A}') = \mathcal{L}_{=\frac{1}{2}}(\mathcal{A}) \setminus \{\varepsilon\}$. En déduire que le problème du vide du langage $\mathcal{L}_{=\frac{1}{2}}(\mathcal{A})$ associé à un automate probabiliste rationnel $\mathcal{A}$ est indécidable.
13. Conclure à l'indécidabilité du vide des langages probabilistes rationnels.

Graphe d'intervalles

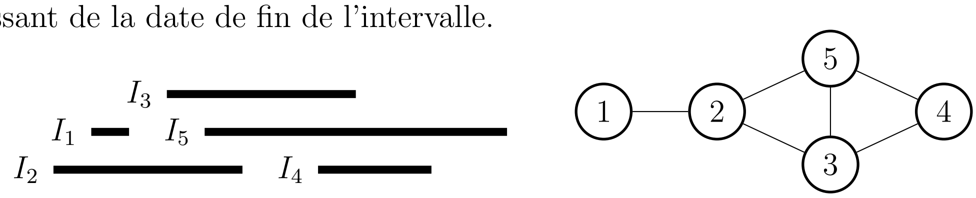
Un **graphe non orienté** G est un couple (V, E) où V est un ensemble de sommets et E un ensemble d'arêtes. Notons $n = |V|$, $m = |E|$. Les sommets sont numérotés de 1 à n. Les sommets u et v sont dits **voisins** s'il y a une arête entre u et v.

Un graphe est un **graphe d'intervalles** si

1. chaque sommet v est associé à un intervalle I_v sur une droite réelle ayant une date de début d_v et une date de fin f_v ;
2. deux sommets u et v sont reliés par une arête si et seulement si les intervalles correspondant à u et v s'intersectent ($I_u \cap I_v \neq \emptyset$).

Par la suite, nous supposons que toutes les dates de début et de fin sont différentes.

Voici un exemple de graphes d'intervalle dont la numérotation des sommets suit l'ordre croissant de la date de fin de l'intervalle.



Problème de clique. On dit qu'un sous-ensemble de sommets $K \subset V$ d'un graphe $G = (V, E)$ est une **clique** s'il existe une arête entre toute paire de sommets de K dans G. La **taille** de K est le nombre de sommets de K. Une clique K est **maximum** s'il n'existe pas de clique K' de taille strictement supérieure à celle de K.

Question 1. Proposer un algorithme calculant une clique maximum en $O(n \log n)$ opérations, dans un graphe d'intervalles représenté par les intervalles.

Un **ensemble indépendant** S de G est un ensemble de sommets deux à deux non voisins dans G. La taille d'un ensemble indépendant est égale au nombre de sommets qu'il contient. Nous cherchons un ensemble indépendant de G de taille maximum. Par la suite, nous notons S^* une solution optimale (un ensemble indépendant de taille maximum).

Considérons l'algorithme glouton suivant, dépendant d'un ordre $\mathcal{T}$ des sommets :

Algorithme 1 : Glouton(G)

Données : Un graphe d'intervalles $G = (V, E)$
Résultat : Un sous-ensemble de sommets
$S \leftarrow \emptyset$; $Q \leftarrow V$;
tant que ($Q \neq \emptyset$) **faire**
  Choisir un sommet v de Q minimal dans l'ordre $\mathcal{T}$;
  si $\{v\} \cup S$ *reste un ensemble indépendant de* G **alors**
    $S \leftarrow \{v\} \cup S$;
    Supprimer les voisins de v dans Q ;
  fin
fin
Retourner S

Question 2. Donner un exemple où l'algorithme ne retourne pas la solution optimale

1. si l'ordre $\mathcal{T}$ sur les sommets est l'ordre croissant des dates de début d_v ;
2. si l'ordre $\mathcal{T}$ sur les sommets est l'ordre croissant sur la longueur des intervalles $f_v - d_v$.

Question 3. Soit S la solution retournée par l'algorithme si l'ordre $\mathcal{T}$ sur les sommets est l'ordre croissant sur la longueur des intervalles $f_v - d_v$.

1. Montrer que chaque sommet j de S est voisin d'au plus deux sommets de S^*.
2. Montrer que $|S^*| \leq 2|S|$.

Notons j le sommet de G telle que sa date de fin f_j soit la plus petite possible.

Question 4. Montrer qu'il existe une solution optimale qui contient le sommet j.

Question 5. Montrer que l'algorithme retourne une solution optimale si l'ordre $\mathcal{T}$ sur les sommets est l'ordre croissant de la date de fin.

Question 6. Simplifier l'algorithme glouton si l'ordre $\mathcal{T}$ sur les sommets est l'ordre croissant de date de fin en supposant donné le graphe par les tableaux de dates de début et de fin. Évaluer sa complexité.

Nous associons une fonction de poids positif sur chaque sommet $w : V \to \mathbb{N}$. Nous cherchons un sous-ensemble indépendant S qui maximise la somme $\sum_{u \in S} w(u)$ de tous les poids de ces sommets. Nous notons S^* un ensemble indépendant de G qui maximise cette quantité.

Convention : À partir de maintenant, nous supposons que les sommets sont numérotés de 1 à n en fonction de leur date de fin f_v : $i < j$ signifie que $f_i < f_j$.

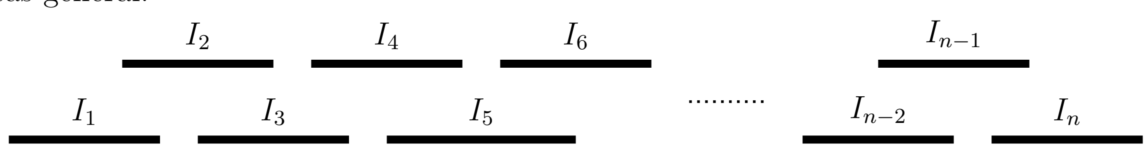
Notons $p(j)$ le plus grand sommet dont l'intervalle finit avant d_j. S'il n'en existe pas, nous supposons que $p(j) = 0$. Dans l'exemple, $p(1) = p(2) = 0$, $p(3) = p(5) = 1$ et $p(4) = 2$.

Question 7. Donner un algorithme en $O(n \log n)$ opérations qui calcule la fonction p pour tous les sommets.

Question 8. Montrer que si $j \in S^*$, alors $S^* \cap \{1, \ldots, p(j)\}$ est un ensemble indépendant de G qui maximise la somme de tous les poids de ces sommets dans le sous-graphe de G ayant les sommets 1 à $p(j)$.

Question 9. Notons $OPT(j)$ le poids d'une solution optimale dans le sous-graphe ayant les sommets 1 à j. Donner une formule de récurrence pour $OPT(j)$

Question 10. Écrire un algorithme récursif qui calcule $OPT(n)$. Appliquer l'algorithme récursif sur l'instance de la figure suivante. Évaluer la complexité de cet algorithme dans le cas général.



Question 11. Comment réduire la complexité de l'algorithme précédent ? Donner sa complexité après amélioration.