

Devoir à la maison

Informatique Fondamentale 1, ENS Rennes

Amaury Pouly

A rendre le **7 novembre 2025**

Toute réponse doit être argumentée. La qualité de la rédaction sera prise en compte dans la notation. Les exercices sont indépendants. Les notations du cours, utilisées dans ce sujet, n'ont pas besoin d'être redéfinies.

1 Calculabilité

On fixe un alphabet Σ contenant au moins deux lettres **a** et **b**. Pour chacun des langages suivants sur Σ , indiquer s'il est décidable ou non, et si ce n'est pas le cas s'il est semi-décidable ou de complémentaire semi-décidable, ou aucun des deux.

1. $\mathcal{L}_1 = \{\mathbf{a}^n : n \in \mathbb{N} \text{ et il existe une machine de Turing } \mathcal{M} \text{ qui accepte exactement } n \text{ mots}\}$.
2. $\mathcal{L}_2 = \{\langle [\mathcal{M}], \mathbf{a}^n \rangle : n \in \mathbb{N} \text{ et } \mathcal{M} \text{ est une machine de Turing qui accepte au moins } n \text{ mots}\}$.
3. $\mathcal{L}_3 = \{\langle [\mathcal{M}], \mathbf{a}^n \rangle : n \in \mathbb{N} \text{ et } \mathcal{M} \text{ est une machine de Turing qui accepte au moins } n \text{ mots en au plus } n \text{ étapes}^1\}$.
4. $\mathcal{L}_4 = \{[\mathcal{M}] : \mathcal{M} \text{ est une machine de Turing qui accepte au plus } n(\mathcal{M}) \text{ mots}\}$ où on pose $n(\mathcal{M}) = 0$ si $\mathcal{M}$ n'accepte aucun mot et $n(\mathcal{M}) = 1 + \min\{|w| : w \in \Sigma^*, \mathcal{M} \text{ accepte } w\}$ sinon.
5. (Difficile) $\mathcal{L}_\infty = \{[\mathcal{M}] : \mathcal{M} \text{ est une machine de Turing qui accepte exactement } |[\mathcal{M}]| \text{ mots}\}$. *Indice : on pourra se servir du théorème 9 du cours, à savoir que pour toute machine de Turing $\mathcal{A}$, il existe une machine $\mathcal{M}$ telle que pour mot w , $\mathcal{M}(w) = \mathcal{A}(\langle [\mathcal{M}], w \rangle)$.*

2 Complexité

On introduit quelques notations concernant les graphes. Un graphe orienté G est la donnée de $G = (V, E)$ où V est un ensemble de sommets et $E \subseteq V^2$ un ensemble d'arêtes (orientées). Pour tout sommet $u \in V$, on pose

$$N_G(u) = \{v \in V : (u, v) \in E\}$$

l'ensemble des voisins (orientés) de u . Etant donné un ensemble $S \subseteq V$ de sommets, on pose :

$$N_G(S) = \bigcup_{u \in S} N_G(u)$$

l'ensemble des sommets accessibles à partir de S en utilisant une arête. Pour tout entier $d \geq 0$, on définit par induction

$$N_G^0(S) = S, \quad N_G^{d+1}(S) = N_G^d(S) \cup N_G(N_G^d(S)), \quad N_G^*(S) = \bigcup_{d \in \mathbb{N}} N_G^d(S).$$

Autrement dit $N_G^*(S)$ (resp. $N_G^d(S)$) est l'ensemble des états accessibles depuis S (resp. accessible en au plus d arêtes).

Dans la suite, et lorsque le graphe G est clair, on pourra simplement écrire $N(S)$, $N^d(S)$ et $N^*(S)$. On pourra aussi écrire « $u \rightarrow v$ » pour dire que $(u, v) \in E$, « $s \rightarrow^d t$ » pour dire que $t \in N_G^d(s)$, et « $s \rightarrow^* t$ » pour dire que $t \in N_G^*(s)$.

Définition 1 (s-t-NON-CONN, aussi appelé *s-t-non-connectivité*).

Entrée : le codage d'un graphe $G = (V, E)$ et deux sommets $s, t \in V$.

Sortie : décider si $t \notin N^*(s)$.

1. Plus formellement, il existe $w_1, \dots, w_n$ des mots distincts tels que pour tout $i = 1, \dots, n$, $\mathcal{M}$ accepte w_i en au plus n étapes de calcul.

On représente le graphe G par la liste de ses sommets suivie de la liste de ses arêtes, par exemple triée par source puis destination. On notera $|G|$ la taille de l'encodage de G . L'encodage exact du graphe G n'est pas important et on pourra **utiliser sans le prouver** le fait qu'on peut représenter un sommet par un mot de taille au plus $O(\log |G|)$ et que pour tous $u, v \in V$ on peut vérifier si $(u, v) \in E$ de façon déterministe en espace $O(\log |G|)$ en supposant que l'encodage de $G = (V, E)$ est disponible sur le ruban d'entrée.

On introduit le lemme suivant **que nous allons prouver plus bas**. Pour cela, nous allons devoir préciser la notion de calcul non déterministe de fonction. Pour toute fonction f , on dit qu'une machine non déterministe $\mathcal{M}$ calcule f si pour tout w :

- $\mathcal{M}$ accepte w sur au moins une branche de calcul, et
- pour toute branche de calcul de $\mathcal{M}(w)$ qui ne rejette pas, la valeur calculée sur le ruban de sortie est $f(w)$.

Lemme 1. *Il existe une machine non déterministe qui sur les entrées $G = (V, E)$ et $v \in V$, calcule $|N_G^{|V|}(v)|$ en espace $O(\log |G|)$.*

Question 1. Montrer que pour graphe $G = (V, E)$ et tout $S \subseteq V$, $N_G^*(S) = N_G^{|V|}(S)$.

Question 2. Montrer qu'étant donné les entrées $G = (V, E)$, $s, t \in V$, et $d \in \mathbb{N}$, on peut décider de façon non déterministe en espace $O(\log |G| + \log d)$ si $t \in N_G^d(s)$. En particulier, on peut décider si $t \in N_G^*(s)$ en espace $O(\log |G|)$

Dans la suite, on appellera **VerifChemin** (G, s, t, d) l'algorithme de la question 2. On notera bien qu'il s'agit d'un algorithme non déterministe qui peut donc accepter ou rejeter selon la branche de calcul ! Autrement dit :

$$\text{VerifChemin}(G, s, t, d) \text{ a une branche qui accepte} \Leftrightarrow t \in N_G^d(s).$$

Question 3. Supposons le Lemme 1 vrai. Montrer qu'étant donné les entrées $G = (V, E)$ et $s, t \in V$, on peut décider de façon non déterministe en espace $O(\log |G|)$ si $t \notin N_G^*(s)$. *Indice : deviner les éléments de $N_G^*(s)$, mais comment s'assurer qu'ils sont distincts ?*

Question 4. En déduire que le Lemme 1 implique que s - t -NON-CONN $\in$ NSPACE($\log n$).

On cherche maintenant à démontrer le Lemme 1 et pour cela on s'intéresse à l'algorithme suivant. On prêtera bien attention au fait que cet algorithme peut soit rejeter, soit retourner une valeur qui est alors soit « oui », soit « non ». En particulier, rejeter n'est pas la même chose que retourner « non ».

Procédure Accessible (G, s, t, k, n)

Entrées : graphe $G = (V, E)$, $s, t \in V$ et $k, n \in \mathbb{N}$

Sorties : si $n = |N_G^k(s)|$ alors calcule « oui » si $t \in N_G^{k+1}(s)$ et « non » si $t \notin N_G^{k+1}(s)$

début

total $\leftarrow 0$

pour tous les $v \in V$ **faire**

$b \leftarrow$ deviner un booléen

si $b = 1$ **alors**

total $\leftarrow$ total + 1

si **VerifChemin** (G, s, v, k) **rejette** **alors**

rejeter

si $v = t$ **ou** $v \rightarrow t$ **alors**

retourner « oui »

si total $\neq n$ **alors**

rejeter

retourner « non »

Question 5. Montrer que **Accessible** (G, s, t, k, n) fonctionne en espace non déterministe $O(\log |G|)$ sur l'entrée $G = (V, E)$, $s, t \in V$, $k \leq |V|$ et $n \leq |V|$.

Question 6. Montrer que pour tout $G = (V, E)$, $s, t \in V$ et $k \in \mathbb{N}$, si $t \in N_G^{k+1}(s)$ alors **Accessible** $(G, s, t, k, |N_G^k(s)|)$ retourne « oui » sur au moins une branche de calcul.

Question 7. Montrer que pour tout $G = (V, E)$, $s, t \in V$ et $k \in \mathbb{N}$, si **Accessible** $(G, s, t, k, |N_G^k(s)|)$ retourne « oui » sur au moins une branche de calcul alors $t \in N_G^{k+1}(s)$.

Question 8. Montrer que pour tout $G = (V, E)$, $s, t \in V$ et $k \in \mathbb{N}$, si $t \notin N_G^{k+1}(s)$ alors **Accessible** $(G, s, t, k, |N_G^k(s)|)$ retourne « non » sur au moins une branche de calcul.

Question 9. Montrer que pour tout $G = (V, E)$, $s, t \in V$ et $k \in \mathbb{N}$, si $\text{Accessible}(G, s, t, k, |N_G^k(s)|)$ retourne « non » sur au moins une branche de calcul alors $t \notin N_G^{k+1}(s)$.

On déduit des questions précédentes que $\text{Accessible}(G, s, t, k, |N_G^k(s)|)$ calcule « $t \in N_G^{k+1}(s)$ » au sens défini plus haut.

Question 10. En déduire un algorithme qui sur l'entrée $G = (V, E)$, $s \in V$, $k \in \mathbb{N}$ et $n = |N_G^k(s)|$, calcule $|N_G^{k+1}(s)|$ de façon non déterministe en espace $O(\log |G|)$.

Question 11. Montrer que le Lemme 1 est vrai.

Question 12. (Difficile) Utiliser le fait que s-t-NON-CONN $\in$ NSPACE($\log n$) pour montrer que NL = co-NL. *Indice : raisonner sur le graphe des configurations.*