

Documents autorisés : notes de cours et de TD.

La notation attachera une grande importance à la clarté et à la concision des justifications.

Le barème est indicatif.

Exercice 1 — Applications directes du cours (5 points). Répondez aux questions suivantes en **justifiant brièvement** vos réponses (toute réponse non justifiée vaut 0 points) :

- On considère le complémentaire du problème de l'arrêt : $\{ \langle M, w \rangle \mid M \text{ ne s'arrête pas sur l'entrée } w \}$.
Ce langage est-il semi-décidable ?
- Donner un exemple de problème qui est dans **NP** mais n'est pas **NP**-complet.
- L'affirmation suivante est-elle vraie ou fausse : "pour tout problème **NL**-complet P et tout problème **NP**-complet Q , P se réduit polynomialement à Q " ?
- On considère le problème **CLIQUE**₄₂. **CLIQUE**₄₂ prend en entrée un graphe non orienté G et demande si G contient une clique (un sous-graphe complet) de taille 42. Quelle est la plus petite classe de complexité contenant **CLIQUE**₄₂ parmi celles que vous connaissez ?
- L'affirmation suivante est-elle vraie ou fausse : "si **P** = **NP** alors tout problème qui est décidable en temps exponentiel est aussi décidable en temps polynomial" ?

Exercice 2 — NP-Complétude (7 points). Dans cet exercice, on suppose que **P** \neq **NP**. On présente quatre problèmes. Pour chacun de ces problèmes dites si il est **NP**-complet ou dans **P**. Attention, les réponses doivent être **prouvées**. Si vous répondez qu'un problème est **NP**-complet vous devez prouver qu'il est dans **NP** et qu'il est **NP**-difficile. Inversement si vous répondez qu'un problème est dans **P** vous devez donner un algorithme polynomial qui le résout.

Remarque : toutes les réductions sont faciles, la principale difficulté est de trouver le bon problème à réduire.

a) SOUS-GRAPHE COMMUN

ENTRÉE : Deux graphes orientés G et H et un entier $k \in \mathbb{N}$.

QUESTION : Existe-t-il un graphe de taille k qui est sous-graphe de G et de H ?

b) FACTEUR COMMUN

ENTRÉE : Un alphabet Σ , deux mots $u, v \in \Sigma^*$ et un entier $n \in \mathbb{N}$.

QUESTION : Existe-t-il un mot $w \in \Sigma^*$ de longueur au moins n qui est facteur de u et v ?
(c'est-à-dire que $u = u_1 w u_2$ et $v = v_1 w v_2$)

c) FORMULE NON-TRIVIALE

ENTRÉE : Une formule de logique propositionnelle φ .

QUESTION : Est-ce que φ et $\neg\varphi$ sont satisfaisables ?

d) VOYAGEUR DE COMMERCE

ENTRÉE : Un graphe orienté $G = (V, E)$, une fonction $f : E \rightarrow \mathbb{N}$ qui associe un poids à chaque arête de G , un sommet $s \in V$, et un entier $max \in \mathbb{N}$.

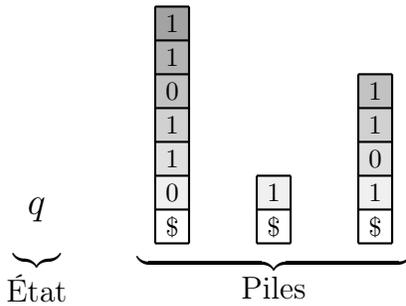
QUESTION : Existe-t-il un chemin dans G qui part et finit dans s , qui visite au moins une fois tous les sommets et tel que la somme des poids des arêtes utilisées est $\leq max$?

Exercice 3 — Machines à Piles (8 points). Dans cet exercice, on présente un nouveau modèle de calcul : les *machines à piles*. Soit $k \geq 1$ un entier, une machine à k piles est un tuple $M = (\Sigma, \Gamma, \$, Q, q_0, q_a, q_r, \delta)$ où,

- Σ est un alphabet fini d'entrée,
- Γ est un alphabet fini de travail qui étend Σ ($\Sigma \subseteq \Gamma$),
- $\$$ est un symbole spécial, appelé symbole de *bas de pile* qui n'appartient pas à Γ (et donc pas à Σ),
- Q est un ensemble fini d'états.
- $q_0 \in Q$ est l'état initial.
- $q_a \in Q$ est l'état final acceptant.
- $q_r \in Q$ est l'état final rejetant.
- $\delta : Q \times (\Gamma \cup \{\$\})^k \rightarrow Q \times (\Gamma \cup \{null\} \cup \{pop\})^k$ est une fonction de transition qui vérifie la condition suivante : si $\delta(q, a_1, \dots, a_k) = (r, b_1, \dots, b_k)$, pour tout i , si $a_i = \$$, alors $b_i \neq pop$.

On va maintenant définir l'exécution d'une machine à piles M quelconque sur un mot d'entrée $w \in \Sigma^*$.

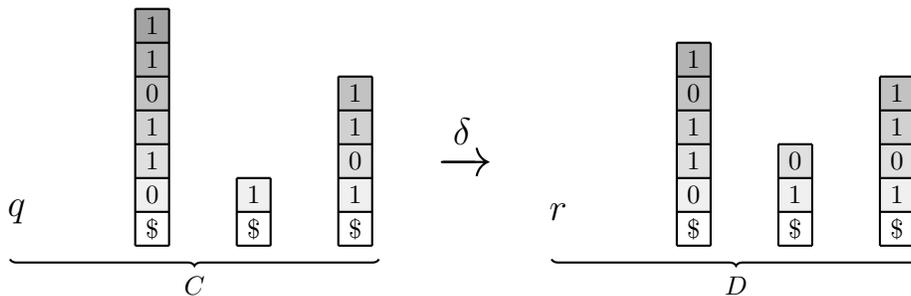
On appelle *configuration* de M un tuple (q, p_1, \dots, p_k) où q est un état dans Q et les p_i sont les piles de M : des mots de la forme $\$v$ où " v " est mot dans Γ^* . On représente souvent une configuration de façon graphique comme dans la figure ci-dessous qui représente la configuration $(q, \$011011, \$1, \$1011)$ d'une machine à 3 piles.



Pour chaque configuration C de M , la fonction de transition δ définit une unique transition de C vers une nouvelle configuration D (on note ce fait $C \xrightarrow{\delta} D$). La configuration D est définie de la façon suivante. On pose $C = (q, p_1, \dots, p_k)$ et pour tout i , on note a_i **la lettre en haut de la pile i** (la dernière lettre du mot p_i). Enfin, on note $(r, b_1, \dots, b_k) = \delta(q, a_1, \dots, a_k)$. La nouvelle configuration D telle que $C \xrightarrow{\delta} D$ est la configuration (r, p'_1, \dots, p'_k) telle que, pour tout i ,

- si $b_i \in \Gamma$ on empile la lettre b_i sur la pile i de C : $p'_i = p_i b_i$.
- si $b_i = null$, on ne modifie pas la pile i de C : $p'_i = p_i$.
- si $b_i = pop$, on dépile la pile i de C : p'_i est p_i privé de sa dernière lettre.

On donne un exemple ci-dessous avec trois piles quand $\delta(q, 1, 1, 1) = (r, pop, 0, null)$.



L'exécution de M sur un mot d'entrée $w \in \Sigma^*$ est maintenant définie de la façon suivante : c'est une suite de configurations C_0, C_1, C_2, \dots (possiblement finie ou infinie, la machine peut ne pas s'arrêter) telles que :

- on part de la configuration $C_0 = (q_0, \$w, \$, \dots, \$)$ (on commence dans l'état initial q_0 et toute les piles sont vides à part la première sur laquelle le mot w est empilé)
- pour tout i , $C_i \xrightarrow{\delta} C_{i+1}$.

L'exécution termine dans une configuration C_n si celle-ci utilise un des deux états finaux (q_a ou q_f). Dans ce cas l'exécution est acceptante si elle termine dans l'état q_a et rejetante si elle termine dans l'état q_f . Le langage d'une machine à pile M est l'ensemble $L(M)$ des mots $w \in \Sigma^*$ tels que l'exécution de M sur w est acceptante.

Partie I : Exemples de Machines à Pile.

- a) Donner une machine à 2 piles M sur l'alphabet d'entrée $\Sigma = \{0, 1\}$ dont le langage $L(M)$ est l'ensemble des mots qui contiennent autant de 0 que de 1.
- b) Donner une machine à 3 piles M sur l'alphabet d'entrée $\Sigma = \{0, 1, 2\}$ dont le langage $L(M)$ est l'ensemble des mots qui contiennent autant de 0 que de 1 que de 2.

Partie II : Machines à Piles et Machines de Turing.

- a) Soit $k \geq 1$ quelconque. Montrer que toute machine à k piles peut être simulée par une machine de Turing. En d'autres termes, on demande d'expliquer comment, à partir du code source d'une machine à k piles, on peut construire une machine de Turing qui accepte le même langage.

Indication : On pourra utiliser des machines de Turing à plusieurs bandes.

- b) Montrer que toute machine de Turing à 1 bande peut être simulée par une machine à 2 piles. En d'autres termes, on demande d'expliquer comment, à partir du code source d'une machine de Turing, on peut construire une machine à 2 piles qui accepte le même langage.
- c) Soit $k \geq 1$ quelconque. Expliquer pourquoi toute machine à k piles peut être simulée par une machine à 2 piles.
- d) Montrer que le problème suivant est indécidable :

ENTRÉE : Une machine M à $k \geq 2$ piles et w un mot d'entrée pour M .
QUESTION : Est-ce que $w \in L(M)$?

Remarque : Le problème est décidable pour les machines à 1 pile (ce résultat est plus difficile à montrer).

■