1 Les mots

1) Étude d'un morphisme de monoïde, mots de Fibonacci

a) Soit A un alphabet. On se donne une application $\alpha : A \to A^*$ et pour tout mot $m = c_1 c_2 \dots c_n \in A^*$ (avec $c_1, \dots, c_n \in A$) on pose $\beta(m) = \alpha(c_1)\alpha(c_2)\dots\alpha(c_n)$. Ainsi $\beta \in \mathscr{F}(A^*, A^*)$.
Montrer que $\forall m, m' \in A^*, \beta(mm') = \beta(m)\beta(m')$

b) On suppose qu'il existe un élément $d \in A$ fixé tel que d soit la première lettre de $\alpha(d)$, c'est–à–dire qu'il existe $v \in A^*$ tel que $\alpha(d) = dv$
On définit une suite de mots (f_n) par $f_0 = d$ et $\forall n \in \mathbb{N}, f_{n+1} = \beta(f_n)$. Montrer que f_n est un préfixe de f_{n+1}.

c) Désormais et pour toute la suite on fixe $A = \{0, 1\}$, $\alpha : \begin{cases} 0 \mapsto 01 \\ 1 \mapsto 0 \end{cases}$ et $d = 0$.
Expliciter f_1, f_2, f_3, f_4.
Montrer que $\forall n \in \mathbb{N}, f_{n+2} = f_{n+1} f_n$.

d) Montrer que si $n \geqslant 1$ alors f_n se termine par $\begin{cases} 01 & \text{si } n \text{ impair} \\ 10 & \text{si } n \text{ pair} \end{cases}$

e) Pour $n \geqslant 1$ soit g_n le préfixe de f_n obtenu en retirant les deux dernières lettres. Montrer que g_n est un palindrome (c'est–à–dire que si on le lit à l'envers on retrouve g_n).

2) Étude de la fermeture réflexive, symétrique et transitive d'une relation

Soit $A = \{a, b\}$.

On définit sur A^* la relation $\mathscr{R}$ par $\forall m, m' \in A^*, m \mathscr{R} m' \iff \exists u, v \in A^*, m = uv$ et $m' = uabv$.

On définit la relation $\mathscr{S}$ par $\forall m, m' \in A^*, m \mathscr{S} m' \iff (m \mathscr{R} m'$ ou $m' \mathscr{R} m)$.

Pour tout entier naturel k on définit par récurrence la relation $\mathscr{S}_k$ par

$$\forall m, m' \in A^*, \begin{cases} m \mathscr{S}_0 m' \iff m = m' \\ m \mathscr{S}_{k+1} m' \iff \exists m'' \in A^* \text{ tel que } m \mathscr{S} m'' \text{ et } m'' \mathscr{S}_k m' \end{cases}$$

Enfin on définit la relation $\mathscr{S}^*$ par $\forall m, m' \in A^*, m \mathscr{S}^* m' \iff \exists k \in \mathbb{N}, m \mathscr{S}_k m'$.

a) Montrer que pour tout mot $m \in A^*$, il existe un couple $(i, j) \in \mathbb{N}^2$ tel que $m \mathscr{S}^* b^i a^j$.

b) Montrer l'unicité de ce couple (i, j).
Indication : pour tout mot w noter w_k le préfixe de w qui contient k lettre, noter $|w_k|_b$ le nombre de caractères b dans w_k et $|w_k|_a$ le nombre de a dans w_k ; puis montrer que pour tout mot w tel que $m \mathscr{S}^* w$, la quantité $\max_k\{|w_k|_b - |w_k|_a\}$ est la même pour w que pour m.

3) Relations d'ordre dans $A^\star$:

On se fixe un alphabet A.

a) Soit u et v deux mots de $A^\star$. On note $u \leqslant_p v$ si u est un préfixe de v.

i) Montrer que $\leqslant_p$ est une relation d'ordre. Est-elle totale ?

ii) Écrire une fonction `plusPetit : string -> string -> int` telle que `plusPetit u v` renvoie 1 si $u \leqslant_p v$ et 0 sinon.

On suppose maintenant que A dispose d'un ordre total noté $\leqslant$.

b) Soit u et v dans $A^\star$ on note $u \leqslant_l v$ si u est un préfixe de v ou s'il existe t, u', v' tels que $u = tu'$, $v = tv'$, $u' \neq \varepsilon$, $v' \neq \varepsilon$ et la première lettre de u' précède la première lettre de v' pour $\leqslant$.

i) Montrer que $\leqslant_l$ est une relation d'ordre. Est-elle totale ?

ii) Écrire une fonction `lexico : string -> string -> int` telle que `lexico u v` renvoie 1 si $u \leqslant_l v$ et 0 sinon.

c) Soit u et v dans $A^\star$ on note $u \prec v$ si $|u| < |v|$ ou si $|u| = |v|$ et $u \leqslant_l v$.

i) Montrer que $\prec$ est une relation d'ordre. Est-elle totale ?

ii) Écrire une fonction `radiciel : string -> string -> int` telle que `radiciel u v` renvoie 1 si $u \prec v$, -1 si $v \prec u$ et 0 sinon.

iii) Montrer que $\prec$ est un ordre bien fondé c'est-à-dire qu'il n'existe pas de suites infinies strictement décroissantes.

4) Distances dans $A^\star$:

On se fixe un alphabet A.

a) Soit u et v deux mots de $A^\star$. On note $\mathsf{plpc}(u, v)$ le plus long préfixe commun de u et v. On pose alors

$$d(u, v) = |u| + |v| - 2|\mathsf{plpc}(u, v)|$$

Vérifier que $d : (A^\star)^2 \to \mathbb{Z}$ est une distance c'est-à-dire que

- $\forall (u, v) \in (A^\star)^2, d(u, v) \geqslant 0$
- $\forall (u, v) \in (A^\star)^2, d(u, v) = 0 \iff u = v$
- $\forall (u, v) \in (A^\star)^2, d(u, v) = d(v, u)$
- $\forall (u, v, w) \in (A^\star)^3, d(u, w) \leqslant d(u, v) + d(v, w)$.

b) On appelle opération élémentaire la suppression d'une lettre dans un mot, l'ajout d'une lettre dans un mot, le remplacement d'une lettre par une autre. Soit u et v dans $A^\star$ on appelle distance d'édition entre u et v et on note $\delta(u, v)$ le plus petit nombre d'opérations pour passer de u à v. Si $u =$ `chien` et $v =$ `niche` on a $\delta(u, v) = 4$.

i) Vérifier que δ est une distance.

ii) Écrire une fonction `distance : string -> string -> int` telle que `distance u v` renvoie la distance de u à v. On mettra en place une méthode de programmation dynamique en calculant pour tout $i \in [\![1, |u|]\!]$ et $j \in [\![1, |v|]\!]$ la distance entre le prefixe de longueur i de u et le préfixe de longueur j de v.

5) a) Combien un mot de longueur n dont toutes les lettres sont distinctes possède-t-il de préfixes ? de suffixes ? de facteurs ?

   b) Combien de facteurs distincts possède le mot $a^m b^n$?

6) Lemme de Lévi

   Soit A un alphabet et soient $u, v, x, y \in A^*$.

   a) Montrer que $uv = xy$ si et seulement si il existe $t \in A^*$ tel que $\Big((ut = x \text{ et } v = ty)$ ou $(u = xt \text{ et } tv = y)\Big)$

   b) En déduire que u et v commutent si et seulement s'il existe $w \in A^*$ et $p, q \in \mathbb{N}$ tels que $u = w^p$ et $v = w^q$.

2 Les langages

7) Lemme d'Arden

   Soient deux langages L et M. On s'intéresse à l'équation $X = L.X + M$ où l'inconnue X est un langage.

   a) Vérifier que $X = L^*.M$ est solution de cette équation.

   b) Montrer que si X est solution alors $L^*.M \subset X$.

   c) Supposons de plus que $\epsilon \notin L$. Montrer qu'alors la seule solution est $X = L^*.M$
   Indication : raisonner par l'absurde et considérer un mot $u \in X \setminus L^*.M$ de longueur minimum.

   d) Dans le cas où $\epsilon \in L$, montrer que tout langage de la forme $L^*.N$ où $M \subset N$ est solution de l'équation $X = L.X + M$

   e) Résoudre l'équation $X = X.L + M$ dans le cas où $\epsilon \notin L$.

8) Anagramme

   Nous décrirons un mot par une liste de caractères sur l'alphabet $\{a, b, \ldots, z\}$. Soit m un mot, un anagramme de m est un mot obtenu par permutation des lettres de m.

   a) En supposant qu'il faille $10^{-6}s$ pour afficher un caractère, combien de temps faut–il à un ordinateur pour afficher tous les anagrammes d'un mot de 10 lettres distinctes ? D'un mot de 20 lettres ?

   b) Écrire une fonction `detecteAnagramme : string -> string -> bool` telle que : `anagramme m n` permet de savoir si n est un anagramme de m.

   c) Écrire une fonction `produitAnagrammes : string -> string list` qui génère la liste de tous les anagrammes d'un mot : on ne cherchera pas à détecter les anagrammes égaux au cas où la même lettre apparaîtrait plusieurs fois, donc pour un mot de longueur n on génère $n!$ anagrammes.

9) Centrale 2003 : langage des parenthésages

Soit l'alphabet $A = \{a, b\}$. On définit par récurrence les langages L_n par $L_0 = \{\varepsilon\}$ et $\forall n \in \mathbb{N}, L_{n+1} = L_n \cup L_n^2 \cup aL_nb$.

Enfin on pose $\mathscr{L}_p = \bigcup_{n\in\mathbb{N}} L_n$.

a) Montrer soigneusement que $abaabb \in \mathscr{L}_p$.

b) Montrer que pour tout $w \in \mathscr{L}_p$, on a $|w|_a = |w|_b$.

c) Montrer que pour tout $w \in \mathscr{L}_p$, si $w \neq \varepsilon$ alors w commence par un a et finit par un b.

d) Soit $w \in \mathscr{L}_p$. Si $0 \leqslant i \leqslant |w| - 1$ on note w_i la i-ème lettre de w en indexant les lettres à partir de 0 (donc si $|w| = n$ on a $w = w_0 \dots w_{n-1}$). On note aussi $w_{i,j}$ le facteur de w constitué des caractères d'indice compris entre i et j : $w_{i,j} = w_i \dots w_j$.
Montrer que pour tout $i \in [\![0, |w|-1]\!]$ tel que $w_i = a$, il existe $j > i$ tel que $w_{i,j} \in \mathscr{L}_p$.
Un tel j est–il unique ?

e) Montrer soigneusement que $w \in \mathscr{L}_p$ si et seulement si $|w|_a = |w|_b$ et, pour tout préfixe u de w, on a $|u|_a \geqslant |u|_b$.

f) Écrire une fonction de signature `string` $\rightarrow$ `bool` qui permet de déterminer si un mot de A^* appartient à $\mathscr{L}_p$ ou non. La complexité de cette fonction devra être linéaire.

g) i) Reprenons le résultat de la question (d) : soit $m \in \mathscr{L}_p$ un mot de longueur n. Pour tout $i \in [\![0, n-1]\!]$, si $m_i = a$ on définit $\phi(i)$ le plus petit entier $j > i$ tel que $m_{i,j} \in \mathscr{L}_p$, et si $m_i = b$ on définit $\phi(i)$ l'entier $k (< i)$ tel que $m_k = a$ et $\phi(k) = i$. Écrire une fonction de signature `string -> int array` qui, pour un mot m de taille n, calcule le tableau $[|\phi(0), \dots, \phi(n-1)|]$.
Quelle est sa complexité ?

ii) Pour améliorer la complexité de la fonction précédente, on va utiliser une structure de pile (LIFO : le dernier élément empilé est le premier dépilé).
Créer :
- un type polymorphe `'a pile`
- une fonction de création de pile vide `creer_pile : unit -> 'a pile`
- une fonction `empiler : 'a -> 'a pile -> unit` (cette fonction est à effet de bord c'est–à–dire qu'elle modifie la pile)
- une fonction `est_vide : 'a pile -> bool` qui permet de tester si une pile est vide
- une fonction `depiler : 'a pile -> 'a` qui renvoie l'élément au sommet de la pile, et qui de plus modifie la pile en supprimant cet élément.

Grâce à ces méthodes de gestion de pile, écrire une fonction de complexité linéaire qui permet de résoudre le problème précédent.

h) Soit c_p le nombre de mots de taille p de $\mathscr{L}_p$. Donner une relation de récurrence vérifiée par c_p.

i) Montrer que la série entière (dite série génératrice des u_n) $\sum (u_n x^n)_{n\geqslant 0}$, avec $u_n = c_{2n}$ a un rayon de convergence strictement positif et que sa somme (appelée fonction génératrice des u_n) vérifie une équation du second degré. Résoudre cette équation et en déduire u_n en fonction de n.

10) Soient deux langages L et M sur un même alphabet. Montrer que $(LM)^*L = L(ML)^*$

11) Soit l'alphabet $A = \{0,1\}$. Pour tout $x \in A$ on définit son conjugué $\overline{x}$ par $\begin{cases} \overline{0} = 1 \\ \overline{1} = 0 \end{cases}$

    On étend la notion de conjugué aux mots :

    Pour tout mot $m = c_0 \dots c_{n-1} \in A^*$ (avec $c_0, \dots, c_{n-1} \in A$) on note $\overline{m} = \overline{c_0} \dots \overline{c_{n-1}}$ (on remarquera que pour tous mots m et m' on a $\overline{(mm')} = \overline{m}\overline{m'}$).

    On considère le morphisme de monoïde $\sigma : A^* \to A^*$ défini par $\sigma(0) = 01$, $\sigma(1) = 10$ et pour tous mots $m, m' \in A^*, \sigma(mm') = \sigma(m)\sigma(m')$.

    On définit alors par récurrence la suite de mots (m_n) par $m_0 = 0$ et $m_{n+1} = \sigma(m_n)$.

    a) Montrer que pour tout mot $m \in A^*, \sigma(\overline{m}) = \overline{\sigma(m)}$.
    b) Exprimer m_{n+1} à l'aide de m_n. En déduire que m_n est un préfixe de m_{n+1}.
    c) Un mot appartenant à A^* sera codé par une liste d'entiers. Programmer alors la fonction `sigma`, réalisation de σ, puis la fonction `itere` telle que `itere n` calcule m_n.
    d) On définit alors la suite $m_\infty \in A^{\mathbb{N}}$ dont tout préfixe est un préfixe d'un m_n. Écrire une fonction qui calcule le n-ième terme de la suite m_∞. Il existe une solution simple de complexité $\mathcal{O}(\log n)$ et de complexité spaciale constante qui n'utilise pas la fonction `itere`.

3 Langages rationnels

12) Projection d'un langage

    Soient A et B deux alphabets disjoints. Pour tout $u \in (A \cup B)^*$ on note $\pi_A(u)$ le mot de A^* obtenu en effaçant toutes les occurrences des lettres de B dans u (si $u \in B^*$ on posera $\pi_A(u) = \varepsilon$).

    Soit L un langage sur $A \cup B$, on pose $\pi_A(L) = \{\pi_A(u) | u \in L\}$.

    Montrer par induction structurelle que si L est rationnel alors $\pi_A(L)$ est rationnel. Si L est décrit par une expression rationnelle, comment obtenir une expression rationnelle qui décrit $\pi_A(L)$?

13) a) Si A est fini montrer que A^* est dénombrable. En déduire qu'il existe des langages non rationnels.
    b) Si $A = \{0,1\}$ construire une énumération de A^*

14) Soit A un alphabet (fini), soit $m = a_1 a_2 \dots a_n \in A^*$ (avec $a_1, \dots, a_n \in A$), on définit son miroir $\tilde{m} = a_n \dots a_1$. Pour tout langage L on définit son miroir $\tilde{L} = \{\tilde{m} | m \in L\}$. Montrer que si L est rationnel alors $\tilde{L}$ est rationnel. Si L est décrit par une expression rationnelle, comment obtenir une expression rationnelle qui décrit $\tilde{L}$?

15) Soit l'alphabet $A = \{a, b\}$. Pour chacun des langages de A^* ci-dessous, donner une expression rationnelle qui le décrit.

    a) L'ensemble des mots se terminant par a
    b) $\{a^n b^m \mid n, m \in \mathbb{N}\}$
    c) Le complémentaire de $\{a^n \mid n \in \mathbb{N}^*\}$
    d) L'ensemble des mots dont la longueur est un multiple de 3
    e) L'ensemble des mots dont le nombre de a est un multiple de 3