Devoir à faire pour le Lundi 2 Mars

Exercice 1 (Mots bien parenthésés ou mots de Dyck)

Σ désigne l'alphabet { (,) }.

Définissons récursivement $\mathcal{M}$ l'ensemble des mots sur l'alphabet Σ de la manière suivante :

— $\epsilon \in \mathcal{M}$ (ϵ désigne le mot vide) et $(\forall x \in \Sigma)(x \in \mathcal{M})$,

— $(\forall x \in \Sigma)(\forall m \in \mathcal{M})(xm \in \mathcal{M})$.

Définissons récursivement $\mathcal{M}^*$ l'ensemble des **mots bien parenthésés** sur l'alphabet Σ de la manière suivante :

— $\epsilon \in \mathcal{M}^*$ (ϵ désigne le mot vide)

— $(\forall m \in \mathcal{M}^*)((m) \in \mathcal{M}^*)$,

— $(\forall m_1 \in \mathcal{M}^*)(\forall m_2 \in \mathcal{M}^*)(m_1 m_2 \in \mathcal{M}^*)$.

Un mot m sera défini en OCaml comme un tableau de caractères :

$$let\ m = [|`(`;`(`;`)`;`(`;`)`;`)`;`)`;`(`;`(`;`)`|];;$$

1. Condition nécessaire.

   Écrire en OCaml, une fonction `parentheses` de signature `char array -> bool` qui permet de savoir si un mot m contient autant de parenthèses ouvrantes que de parenthèses fermantes.

   Cette information, sur le nombre de parenthèses ouvrantes et le nombre de parenthèses fermantes d'un mot m, suffit-elle pour savoir si le mot m est bien parenthésé ? Justifier votre réponse.

2. Condition nécessaire et suffisante : profil d'un mot.

   Étant donné un mot m, notons $l(m)$ sa longueur, définissons alors le **profil** de m comme l'application p_m définie sur $[\![0\,,\,l(m)]\!]$ à valeurs dans $\mathbb{Z}$ par :

$$\begin{cases} p_m(0) = 0 \\ \forall i \in [\![1\,,\,l(m)]\!] \mapsto p_m(i) = p_m(i-1) + \begin{cases} +1 \text{ si } m.(i-1) = `(` \\ -1 \text{ si } m.(i-1) = `)` \end{cases} \end{cases}.$$

   (a) Calculer le profil du mot (( ) ( ))) (( ). Ce mot est-il bien parenthésé ?

   (b) Écrire en OCaml une fonction `profil` de signature `char array -> int array` qui calcule le profil d'un mot m.

   (c) Soit m un mot différent du mot vide tel que : $\begin{cases} p_m(l(m)) = 0 \\ p_m \text{ est à valeurs dans } \mathbb{N} \end{cases}.$

   — Justifier l'existence d'un entier k tel que :

$$k = \min\{i \in [\![1\,,\,l(m)]\!] / p_m(i) = 0\}.$$

   — On suppose que $k = l(m)$.

   Montrer qu'il existe un mot m' de longueur $l(m) - 2$ tel que : $m = (m')$.

   Définir le profil de m' en fonction du profil de m et montrer qu'il vérifie :

$$\begin{cases} p_{m'}(l(m')) = 0 \\ p_{m'} \text{ est à valeurs dans } \mathbb{N} \end{cases}.$$

   — On suppose que : $k < l(m)$.

   Montrer qu'il existe deux mots m_1 et m_2 de longueur strictement inférieure à $l(m)$ tels que : $m = m_1 m_2$.

Définir le profil de m_1 et le profil de m_2 en fonction du profil de m et montrer qu'ils vérifient :

$$\left\{\begin{array}{l} p_{m_1}(l(m_1)) = 0 \\ p_{m_1} \text{ est à valeurs dans } \mathbb{N} \end{array}\right. \text{ et } \left\{\begin{array}{l} p_{m_2}(l(m_2)) = 0 \\ p_{m_2} \text{ est à valeurs dans } \mathbb{N} \end{array}\right. .$$

— Montrer que m est bien parenthésé.

(d) Soit m un mot bien parenthésé.

Montrer que le profil de m vérifie : $\left\{\begin{array}{l} p_m(l(m)) = 0 \\ p_m \text{ est à valeurs dans } \mathbb{N} \end{array}\right. .$

(e) En déduire une condition nécessaire et suffisante pour qu'un mot soit bien parenthésé.

(f) Écrire en OCaml une fonction `bien_parenthese` de signature `char array -> bool` qui permet de savoir si un mot m est bien parenthésé.