Centrale Informatique MP 2003 — Corrigé

Ce corrigé est proposé par Samuel Mimram (ENS Lyon) ; il a été relu par Jean-Baptiste Rouquier (ENS Lyon) et Vincent Puyhaubert (Professeur en CPGE).

Le sujet traite du problème classique qui consiste à déterminer si un mot est bien parenthésé : étant donné une suite de parenthèses, on cherche à savoir si chaque parenthèse fermante correspond à une parenthèse préalablement ouverte et si toutes les parenthèses ouvrantes sont bien fermées. Il comporte trois parties, qui ne sont pas indépendantes, dont les questions sont de difficulté progressive :

- la première partie établit quelques propriétés des mots bien parenthésés, qui serviront par la suite ;
- la deuxième présente différents algorithmes qui permettent de trouver la parenthèse ouvrante associée à une parenthèse fermante (et vice versa) dans un mot bien parenthésé ;
- la troisième traite d'une extension du problème dans laquelle les mots sont constitués de parenthèses et de crochets.

Ce sujet contient quelques questions difficiles, en particulier l'algorithme de la question II.B.6 (qui requiert de l'intuition), et demande d'être précis et méticuleux. Le raisonnement par récurrence forte est abondamment utilisé et des connaissances de cours sur les langages, les expressions régulières et les automates sont nécessaires. Des implémentations de la plupart des algorithmes proposés sont demandées. Ce problème permet ainsi de réviser certaines techniques et propriétés fondamentales pour l'épreuve d'informatique des concours.

Indications

I.A.1 Montrer que le mot appartient à L_3 en utilisant la définition des L_n (en particulier montrer que $aabb \in L_2$).

I.A.2 Montrer la propriété sur les L_n par récurrence sur n.

I.A.3 Montrer la propriété sur les L_n par récurrence sur n. Pour le contre-exemple, considérer le mot $abaabb$.

I.B.1 Utiliser le lemme de l'étoile.

I.B.2 Construire l'automate produit de deux automates.

I.B.3 Utiliser le langage défini par l'expression régulière a^*b^*.

I.C.1 Montrer le résultat par double implication. Le sens direct se montre sur les L_n par récurrence sur n, l'autre sens par récurrence sur la longueur de w.

I.C.2 Lire w de gauche à droite et stocker le nombre de a et le nombre de b rencontrés depuis le début. Utiliser la caractérisation de la question précédente, pour conclure.

II.B.1 Montrer par l'absurde l'existence d'un représentant irréductible, en construisant une suite infinie de mots de longueurs décroissantes, puis montrer son unicité en « coloriant » les lettres ajoutées au mot irréductible de départ.

II.B.2 Enlever le plus possible de facteurs () au concaténé de w_1 et w_2.

II.B.3 Montrer qu'un mot est bien parenthésé si et seulement si son représentant irréductible est ε en prouvant les deux sens de l'équivalence par récurrence.

II.B.4 Déterminer le nombre de nœuds de l'arbre réduit en fonction du nombre de ses feuilles.

II.B.6 Faire un algorithme en deux étapes. La première vise à trouver un sous-mot de w commençant par la parenthèse ouvrante qui contient la parenthèse fermante correspondante. Ceci se fait en ajoutant des sous-arbres à une liste F de sous-arbres de l'arbre réduit jusqu'à ce que le représentant irréductible du mot $m(\mathrm{F})$ formé par les feuilles des arbres de F ne soit plus de la forme $(^i$. On choisira F de façon à ce que ses éléments soient « consécutifs », c'est-à-dire que l'ensemble de toutes leurs feuilles forment un facteur de w. Dans une seconde étape, réduire le dernier sous-arbre de F (en enlevant des feuilles « à droite »), de sorte que le représentant irréductible de $m(\mathrm{F})$ soit ε.

II.B.8 Il faut calculer les étages de l'arbre en parallèle, chaque étage étant calculé en temps $\mathrm{O}(1)$.

III.A S'inspirer de la définition de $\mathscr{L}_\mathrm{P}$.

III.B Raisonner comme à la question I.B.3. Pour la fonction, utiliser une pile comme à la question II.A.2 (on ne cherche pas à calculer la parenthèse fermante correspondante à une parenthèse ouvrante mais juste à vérifier qu'une parenthèse ouvrante n'est pas fermée par un crochet et vice versa).

III.C Adapter la fonction de la question II.A.2.

Même si l'énoncé ne le demande pas, il est très facile de fournir une implémentation des piles à l'aide de références sur des listes :

```
type 'a pile == 'a list ref;;

let creer_pile () = ref [];;

let empiler x p = p := x :: !p;;

let depiler p = let t::q = !p in p := q; t;;

let est_vide p = !p = [];;
```

Cela peut vous être utile pour tester vos fonctions.

Caml génère une alerte lors de l'évaluation de la définition de la fonction `depiler` qui correspond au fait qu'on ne gère pas le cas où on essayerait de dépiler un élément d'une pile vide. La gestion de tels cas n'est pas exigible en concours.

Pour les preuves, il faut utiliser la définition des mots bien parenthésés comme union des L_n et non la définition informelle de l'introduction.

I. Mots bien parenthésés

I.A.1 Montrons que le mot *abaabb* est dans $\mathscr{L}_\mathrm{P}$.

Le mot ε est dans L_0 par définition. Or on a $a\mathrm{L}_0 b \subseteq \mathrm{L}_1$, donc $ab = a\varepsilon b$ est dans L_1. Pour les mêmes raisons, $aabb = a(ab)b$ est dans L_2 car ab est dans L_1. Enfin, $\mathrm{L}_1 \subseteq \mathrm{L}_2$, donc $ab \in \mathrm{L}_2$, et ${\mathrm{L}_2}^2 \subseteq \mathrm{L}_3$ donc *abaabb* qui est le concaténé de ab et de $aabb$ est dans L_3. Ainsi,

$$\boxed{abaabb \in \mathscr{L}_\mathrm{P}}$$

I.A.2 Soit $\mathscr{P}(n)$ la propriété « pour tout mot w de L_n, on a $|w|_a = |w|_b$ et si $w \neq \varepsilon$, alors w commence par un a et se termine par un b ». Montrons qu'elle est vraie pour tout $n \in \mathbb{N}$.

- $\underline{\mathscr{P}(0)}$: par définition, $\mathrm{L}_0 = \{\varepsilon\}$ et la propriété est trivialement vérifiée.
- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$: soit $w \in \mathrm{L}_{n+1}$. Par définition de L_{n+1}, il y a trois possibilités (non exclusives).
  - $w \in \mathrm{L}_n$: la propriété est vérifiée par hypothèse de récurrence.
  - $w \in {\mathrm{L}_n}^2$: il existe deux mots w_1 et w_2 dans L_n tels que $w = w_1 w_2$. On peut supposer w_1 et w_2 différents du mot vide, sinon nous sommes dans le cas précédent. Par hypothèse de récurrence, on a $|w_1|_a = |w_1|_b$ et $|w_2|_a = |w_2|_b$ donc

    $$|w|_a = |w_1|_a + |w_2|_a = |w_1|_b + |w_2|_b = |w|_b$$

    Par hypothèse de récurrence, on sait aussi que w_1 et w_2 commencent par a et se terminent par b. Donc w commence lui aussi par a et se termine par b.

– $w \in a\mathrm{L}_n b$: il existe un mot w' de L_n tel que $w = aw'b$. Par hypothèse de récurrence, on sait que $|w'|_a = |w'|_b$. On en déduit

$$|w|_a = |w'|_a + 1 = |w'|_b + 1 = |w|_b$$

Il est de plus immédiat de constater que w est distinct de ε, commence par a et se termine par b.

La propriété $\mathscr{P}(n)$ est donc vérifiée pour tout entier n.

Soit w un mot appartenant à $\mathscr{L}_\mathrm{P} = \bigcup\limits_{n\in\mathbb{N}} \mathrm{L}_n$. Il existe un entier n tel que w soit dans L_n. En appliquant la propriété précédente, on en déduit

$$\boxed{|w|_a = |w|_b \text{ et si } w \neq \varepsilon \text{ alors } w \text{ commence par un } a \text{ et se termine par un } b.}$$

> Le langage $\mathscr{L}_\mathrm{P}$ est défini comme l'union des langages L_n. Pour montrer une propriété sur un langage défini de la sorte, il faut penser qu'il suffit de montrer la propriété sur tous les langages L_n.

I.A.3 On suppose que l'indice de la première lettre d'un mot est 0 (pour être en accord avec les conventions utilisées par Caml). Soit $\mathscr{P}(n)$ la propriété « pour tout mot w de L_n, pour tout i tel que $w_i = a$, il existe $j > i$ tel que $w[i \dots j] \in \mathscr{L}_\mathrm{P}$ ». Montrons qu'elle est vraie pour tout entier n.

- $\underline{\mathscr{P}(0)}$: on a $\mathrm{L}_0 = \{\varepsilon\}$ et la propriété est trivialement vérifiée car si $w \in \mathrm{L}_0$ alors $w = \varepsilon$ et il n'y a aucun entier i tel que $w_i = a$.
- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$: soient w un mot de L_{n+1} et i un entier tel que $w_i = a$. Par définition de L_{n+1}, il y a trois possibilités (non exclusives).
  - $w \in \mathrm{L}_n$: la propriété est vérifiée par hypothèse de récurrence.
  - $w \in {\mathrm{L}_n}^2$: il existe deux mots w_1 et w_2, que l'on peut supposer différents de ε, dans L_n, tels que $w = w_1 w_2$. Si $i < |w_1|$ (w_i est une lettre de w_1), alors, par hypothèse de récurrence, il existe $j > i$ tel que $w_1[i \dots j]$ soit dans $\mathscr{L}_\mathrm{P}$. Dans ce cas, $w[i \dots j] = w_1[i \dots j]$ est un mot de $\mathscr{L}_\mathrm{P}$. Sinon, on a $i \geqslant |w_1|$ et w_i est une lettre de w_2 d'indice $i - |w_1|$ dans w_2. Par hypothèse de récurrence, il existe $j' > i - |w_1|$ tel que $w_2[i - |w_1| \dots j']$ soit dans $\mathscr{L}_\mathrm{P}$ et, en posant $j = j' + |w_1|$, on a $w[i \dots j] = w_2[i - |w_1| \dots j']$ qui est un mot de $\mathscr{L}_p$.
  - $w \in a\mathrm{L}_n b$: il existe un mot w' de L_n tel que $w = aw'b$. Si $i = 0$ alors $j = |w| - 1$ convient, car le mot $w[i \dots j]$, qui est égal à w, est dans L_{n+1} donc dans $\mathscr{L}_\mathrm{P}$. On ne peut avoir $i = |w| - 1$, car $w_{|w|-1} = b$. Sinon, on a $0 < i < |w| - 1$ et w_i est une lettre de w' d'indice $i - 1$ dans w'. Par hypothèse de récurrence, on sait qu'il existe $j' > i - 1$ tel que $w'[i - 1 \dots j']$ soit dans $\mathscr{L}_\mathrm{P}$. Ce mot est aussi un sous-mot de w car, en posant $j = j' + 1$, on a $w[i \dots j] = w'[i - 1 \dots j']$.

La propriété $\mathscr{P}(n)$ est donc vérifiée pour tout entier n.

Soient w un mot appartenant à $\mathscr{L}_\mathrm{P} = \bigcup\limits_{n\in\mathbb{N}} \mathrm{L}_n$ et i un entier tel que $w_i = a$. Il existe un entier n tel que w soit dans L_n. En appliquant la propriété précédente, on en déduit :

$$\boxed{\text{Il existe } j > i \text{ tel que } w[i \dots j] \text{ soit dans } \mathscr{L}_\mathrm{P}.}$$