

# TP 2 - Recherche de motif et barman aveugle

7 octobre 2011

Pour toutes questions, suggestions, remarques ou autres, n'hésitez pas à m'envoyer un mail à edbonnet@hotmail.com en mettant en objet [TP Caml].

## 1 Introduction

Les automates sont des machines simples. Comme vous le présentez sans doute et verrez dans la suite du cours, leur expressivité de calcul (pour le dire informellement) est limitée. Cependant, ils surgissent dans de nombreuses situations algorithmiques, et nous allons étudier dans la suite deux exemples.

## 2 Recherche de motif

La recherche de motif a des applications directes évidentes : éditeurs de texte, moteurs de recherche, ainsi que des applications dans des domaines variés allant de la génétique à la musique. Nous allons voir ici que la structure d'automate peut accélérer la recherche et conduire à l'optimalité.

Pour alléger le code, on pourra travailler avec un alphabet  $\{0, 1, \dots, \sigma - 1\}$  de type int, et donc avec des motifs et textes de type int vect. L'objectif de la recherche est de renvoyer toutes les positions d'occurrence du motif dans le texte. Dans la suite  $m$  dénote la taille du motif et  $n$  la taille du texte.

### 2.1 Algorithme de recherche naïf

Il faut toujours commencer par l'algorithme naïf pour valoriser tous les raffinements qui suivent inévitablement.

**Question 1** *Implémenter l'algorithme naïf. Quelle est sa complexité ? Pourquoi n'est-ce pas catastrophique en pratique ? Sur quelles types d'instances l'est-ce ?*

### 2.2 Recherche optimisée par un automate

L'idée principale pour améliorer l'algorithme naïf est de constater que l'on peut parcourir une unique fois le texte en se déplaçant en parallèle dans un automate qui nous indique notre position dans la reconnaissance du motif.

**Question 2** *De quel automate parlons-nous ? Implémenter l'algorithme qui construit l'automate associé au motif. Quelle est sa complexité ? Quelle est maintenant la complexité de la recherche ?*

Il serait maintenant intéressant d'améliorer les performances de la construction de l'automate. On note  $M$  le motif,  $T$  le texte et  $A[1 \dots k]$  correspond au  $k$  premiers caractères de la chaîne  $A$ . Pour vous mettre sur la voie, on note  $\pi(k) := \max\{k' < k \mid M[1 \dots k'] \text{ est un suffixe de } M[1 \dots k]\}$ .

**Question 3** *En quoi la fonction  $\pi$  peut nous servir à calculer l'automate ? Écrire un programme qui prend en entrée le motif et un tableau codant la fonction  $\pi$  et qui calcule l'automate associé.*

On définit  $\pi^*$  par  $\pi^*(0) = \{0\}$  et  $\pi^*(k) = \{\pi(k)\} \cup \pi^*(\pi(k))$ . Remarquons que  $\pi(i) = 1 + \max\{j \mid j \in \pi^*(i-1) \wedge M[j+1] = M[i]\}$ , avec la convention peu habituelle  $\max(\emptyset) = -1$ .

**Question 4** *Implémenter le calcul de la fonction  $\pi$ . Quelle est sa complexité (attention, pour une fois cette question est délicate) ?*

### 2.3 Algorithme de Morris-Pratt

Au delà de l'amusante coïncidence pour les connaisseurs de la série "Urgences", l'algorithme de Morris-Pratt permet de calculer l'automate sans dépendance en la taille de l'alphabet. L'idée est de se servir toujours de  $\pi$  et de généraliser l'automate à une machine capable de faire une transition sans progresser dans la lecture du mot (sorte d' $\epsilon$ -transition forcée).

**Question 5** *Implémenter l'algorithme de Morris-Pratt. Pourquoi la recherche est toujours en  $O(n)$  ?*

Pour être relativement exhaustif sur la recherche de motif, on citera l'algorithme de Knuth-Morris-Pratt (KMP), raffinement de Morris-Pratt par une simplification de l'automate et l'algorithme de Simon qui se base sur une étude plus fine des transitions arrières.

## 3 Barman aveugle

Le jeu du barman aveugle est un jeu stratégique à 2 joueurs (au sens de la théorie des jeux) qui se joue avec 4 verres disposés en carré, placés au début uniformément aléatoirement retournés ou non. Le barman aveugle, qui, comme on aura pu l'anticiper, ne voit pas, choisit entre 3 types d'actions : retourner 1 verre, retourner 2 verres adjacents, retourner 2 verres opposés. L'autre joueur, que l'on pourrait appeler le pilier voyant doit choisir une action respectant le choix du barman. On itère alors le principe et le barman gagne si les 4 verres sont dans le même état.

**Question 6** *Implémenter un algorithme qui donne une stratégie gagnante pour le barman (si elle existe) sur ce jeu et sur toute généralisation raisonnable.*