

Méthodes Empiriques et Langages de Script

TP 3

Gabriele A. Musillo
musillo4@etu.unige.ch

October 24, 2006

1 Exercice 1: les structures de contrôle

- considérez le pseudo-code de la boucle `repeat-until` suivante et simulez-la au moyen d'une boucle PERL `do-while`:

```
REPEAT
  a_statement;
UNTIL ($p == 0)
```

- formulez la conditionnelle équivalente au bloc suivant:

```
$b = 1;
while ($b && $e) {
  a_statement;
  $b = 0;
}
while ($b) {
  another_statement;
  $b = 0;
}
```

- codez un programme qui permet d'afficher des patterns carrés $N \times N$ telles que:

```
+ + + + +
- + + + +
- - + + +
- - - + +
- - - - +
(un pattern 5 × 5)
```

- codez un programme qui permet d'afficher n'importe quelle pattern telle que:

```

- + + + + + -
- - + + + - -
- - - + - - -
- - - - - - -
- - - + - - -
- - + + + - -
- + + + + + -

```

(une pattern de hauteur 3)

Indication: remarquez et exploitez les symétries de cette pattern.

2 Exercice 2: les n -grammes

Un n -gramme est une séquence de n mots. Vous supposerez que 2 mots sont séparés par un ou plusieurs blancs.

Codez un programme qui lit un fichier de texte et un entier n (≥ 2) et qui écrit, sur la sortie standard, les n -grammes ainsi que leur fréquence.

3 Exercice 3: l'entropie de Shannon

L'entropie $h(\sigma)$ d'un symbole σ est définie de la manière suivante:

$$h(\sigma) = -\log_2 P(\sigma)$$

La probabilité $P(\sigma)$ d'un symbole σ est égale à la fréquence relative du symbole σ . La fréquence relative d'un symbole est le rapport de sa fréquence et de la somme des fréquences de tous les symboles.

L'entropie d'un ensemble de symboles \mathcal{A} est définie ainsi:

$$H(\mathcal{A}) = -\sum_{x \in \mathcal{A}} P(x) \log_2 P(x)$$

Codez un programme qui calcule l'entropie h de chaque symbole apparaissant dans un fichier de texte ainsi que l'entropie H de l'ensemble des symboles du fichier.

PERL met à disposition la fonction `log($n)` qui retourne le logarithme de base e du nombre $\$n$. Pour convertir `log($n)` en base 2, il suffit d'appliquer la formule `log_2($n) = log($n)/log(2)`.

4 Exercice 4: Énumération des permutations d'une séquence.

On sait qu'il y a $n!$ permutations d'un ensemble de n éléments. Voici, par exemple, les $3!$ permutations de l'ensemble $\{a, b, c\}$:

abc acb bac bca cab cba

On remarque que l'ensemble des permutations de $\{a, b, c\}$ peut être partitionné en 3 sous-ensembles: le sous-ensemble des permutations qui commencent par un a , le sous-ensemble des permutations qui débutent avec un b et celui des permutations avec un c initial. Remarquez que chacun de ces sous-ensembles est formé d'un caractère initial suivi d'une permutation des caractères distincts du caractère initial. Cette remarque suggère qu'une permutation d'un ensemble ou d'une chaîne de caractères peut être définie récursivement:

une permutation d'une chaîne n'est rien d'autre qu'un caractère c de cette chaîne auquel on a concaténé une permutation des caractères distincts de c .

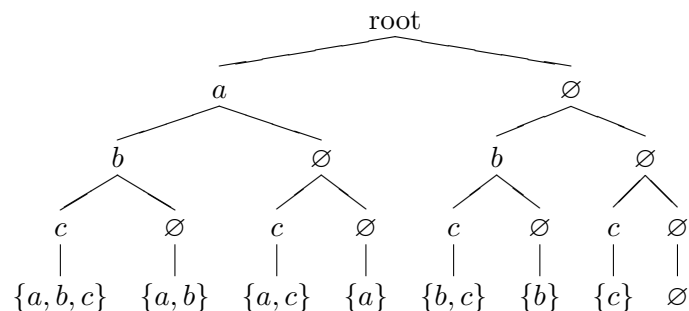
Sur la base de cette définition, implémentez une sous-routine récursive qui énumère sur la sortie standard toutes les permutations d'une chaîne de caractères passée en paramètre.

5 Exercice 5: Énumération des combinaisons d'un ensemble.

On sait qu'il y a 2^n sous-ensembles d'un ensemble de n éléments. Voici, par exemple, les 2^3 sous-ensembles de $\{a, b, c\}$:

$\{a, b, c\} \{a, b\} \{a, c\} \{a\} \{b, c\} \{b\} \{c\} \{\}$

Chacun de ces sous-ensembles est représenté par un chemin dans l'arbre suivant:



Cet arbre suggère que les sous-ensembles de $\{a, b, c\}$ peuvent être énumérés de manière récursive. En effet, cet arbre montre qu'un sous-ensemble de $\{a, b, c\}$ est ou la réunion de $\{a\}$ et d'un sous-ensemble de $\{b, c\}$ ou un sous-ensemble de $\{b, c\}$.

Sur la base de cette définition récursive, codez une sous-routine récursive qui génère tous les sous-ensembles d'un ensemble passé en paramètre.