CM3 : Données non structurées

3. Données non structurées

TBD : Manu

- chaine de caractères
- fichiers textes (stockage, utf8, lecture et ecriture)
- expressions régulières

3.1 Données texte

3.1.1 Codage des caractères

```
x = 'a'
```

- x une variable de type caractère
- a la valeur numérique (encodé):
  - Codage ASCII :
    - codage de symboles du clavier numérique.
    - Le nombre de symboles étant inférieur à 256, on le code sur un octet : 2^8 (= 256) valeurs différentes
  - Codage UTF-8 :
    - codage universel qui permet entre autre de coder les accents
  - Codage ISO 90...

3.1.2 Codage des mots et du texte

Chaîne de caractère :

```
s = "bonjour"
```

- s est une variable
- `"bonjour"` est une séquence de caractères
  - "chaîne" de caractères : type *string* en anglais
  - assimilable à un tableau de caractère :

```
for c in s :
  print (c)
```

Affiche :

```
b
```

```
o
n
j
o
u
r
```

Le programme affiche les caractère 1 par 1, c'est une "chaîne" de plusieurs caractères individuels.

3.1.3 Textes et bases de textes

Un texte, au même titre qu'un mot, est une chaîne de caractères (dont la longueur est définie par la longueur de la séquence de caractères qui définissent le textes, ponctuations, espaces et caractères de retour à la ligne compris.

Les caractères séparateurs

Par définition les caractères séparateurs définissent la taille des espaces entre les mots, ainsi que les passages à la ligne lors de l'affichage du texte.

- `""` : caractère nul
- `" "` : un espace simple
- `"\t"` : tabulation
- `"\n"` : passage à la ligne ("retour chariot")
- `"\b"` : retour arrière ("*backspace*")
- etc.

- Bases de textes : ensemble constitué de plusieurs textes
  - Bases de documents,
    - Dossiers contenant des documents
    - Collections de livres (électroniques)
    - → 10 - 10^4
  - Contenus en ligne (descriptifs de films, articles de journaux, descriptifs de produits.)
    - → 10^5 - 10^6
  - Ensemble du web (les pages web, en tant que telles, peuvent être considérées comme du texte mis en forme par du html).
    - → 10^9
  - Messageries, Blogs, Forums :
    - 10^3 - 10^6

3.2 Recherche dans les textes

Exemples :

- → Recherche d'un terme : "artichaud"
- → Recherche d'un motif (expression régulière) : une adresse email, une URL, un expéditeur, un

numéro tel

Problématiques de la recherche de texte :

- temps de réponse des algorithmes :
  - l'utilisateur classique veut attendre moins de 2 à 3 secondes.
  - Sur des bases extrêmement grandes (type recherche web), il faut donc être très performant pour atteindre ces temps de réponses.
- Stockage des données :
  - intégralité des textes ou simple descriptif/résumé?
  - Les moteurs de recherche par exemple ne stockent aucune page web, ils ne stockent que des index et des références.

Remarque :

Un document texte pourra être décrit soit comme :

- une séquence de caractères (lettres)
- une séquence de termes (mots)

3.2.1 Recherche simple

TODO

d : document de taille `m` On cherche un algorithme qui retourne toutes les occurrences (position dans le doc) d'un certain terme `t` (de taille k<m)

`t = "ami"`

d :

```
"Les amis de mes amis sont mes amis."
     ^           ^             ^
     4           16            30
```

- → La position de la première occurrence du terme t :
- → Les positions de toutes les occurrences du terme t :
- Complexité : $O(m \times k)$

Remarque : Il peut être nécessaire de vérifier que le terme est précédé et suivi par les caractères d'espacement pour éviter de détecter les mots dont le mot recherché est préfixe ou suffixe.

3.2.2 Compter les mots

TODO

Lecture séquentielle des caractères :

```
"Les amis de mes amis sont mes amis."
 ^
 position initiale de la tête de lecture
```

- Il ne faut compter que les debuts de mots
- Un début de mot est un caractère *alphanumérique* :

```
{a,à,ä,b,c,ç,d,e,é,è,ê,ë,...,z,A,B,C,...,Z,1,2,3,...,0}
```

- Tout ce qui n'est pas alphanumérique est un caractère *séparateur*

```
{!,#,$,%,&,",',...}
```

- Autrement dit on compte les couples (caractère séparateur, caractère alphanumérique)
  - remarque : le début de chaîne compte comme un caractère séparateur

remarque : pour extraire la liste des mots présents dans le texte, on doit identifier les débuts et les fins de mots :

- un début de mot est un couple (sep., alphanum.)
- une fin de mot est un couple (alphanum., sep.)

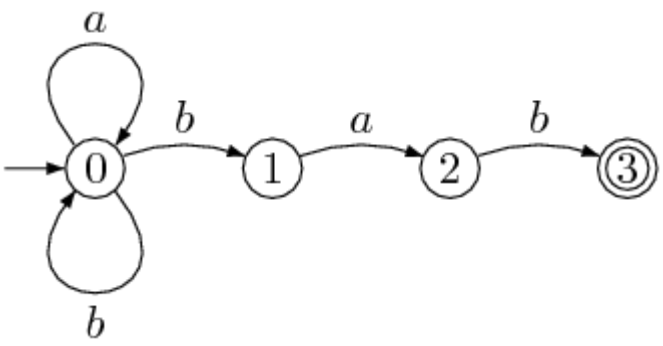
3.2.3 Recherche de motifs et expressions régulières

De manière plus générale recherche d'expressions peut être effectuée à l'aide d'**automates finis**.

- Un automate fini est un graphe défini par des états et des transitions d'état.
- Parmi les états on distingue les états initiaux et terminaux.
- Les transitions permettent de passer d'un état à l'autre selon une condition :
  - Si la condition est remplie l'automate change d'état
  - Si aucune condition n'est remplie, l'automate s'arrête
- Lorsque l'automate s'arrête, on regarde si l'état est terminal. S'il est terminal, le mot est accepté (autrement dit le motif est "reconnu")

Lecture séquentielle des caractères :

```
"Les amis de mes amis sont mes amis."
 ^
 position initiale de la tête de lecture
```



Représentation graphique : (source : `Luc.Maranget@inria.fr`)

- les états dans des cercles,
- l'unique état initial par une flèche pointant sur un état,
- les états terminaux par un double cercle concentrique sur l'état.
- Les transitions d'état quant à elles sont représentées par une flèche allant de l'état de départ jusqu'à l'état d'arrivée indexée par le caractère (ou le groupe de caractères) autorisant la transition.

Traduction sous forme d'expression régulière :

```
(a|b)*ab
```

Ce motif reconnaît les mots suivants :

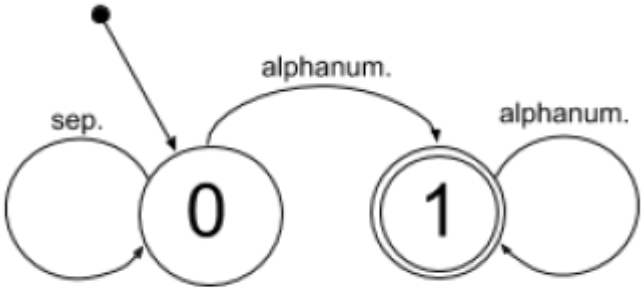
```
ab
aab
bab
aaab
abab
baab
bbab
aaaab
etc..
```

Remarques :

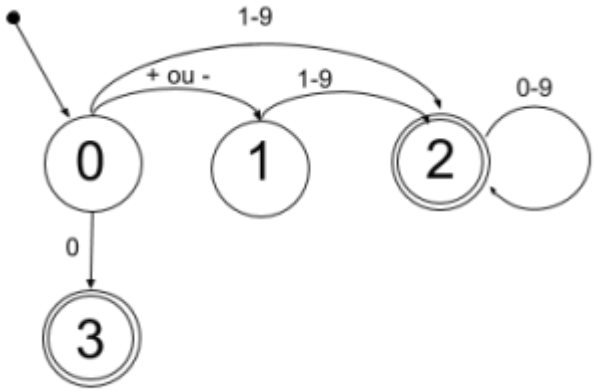
- Les classes d'expressions qui peuvent être reconnues par un automate fini sont appelées des *expressions régulières*
- En python, les expressions régulières s'expriment à l'aide d'une syntaxe spécifique à l'aide de la librairie `re`

Exemples:

- Reconnaître un mot :



- Reconnaître un nombre entier : `+4`, `-455`, `1024`, `0`



- Reconnaître un nombre réel :
  - ??

Syntaxe des expressions régulières Python

Définition : Il s'agit d'une syntaxe "condensée" de description d'automates finis permettant de reconnaître des motifs dans un texte.

En Python, les expressions régulières sont implémentées dans la librairie `re`

```
import re

d = "Les astronautes de la mission Gemini 8 sont désignés le 20 septembre
1965 : Armstrong est le commandant et David Scott le pilote. Ce dernier est
le premier membre du groupe d'astronautes à recevoir une place dans
l'équipage titulaire d'une mission spatiale. La mission est lancée le 16
mars 1966. Celle-ci est la plus complexe réalisée jusque-là, avec un rendez-
vous et un amarrage du vaisseau Gemini avec l'étage de fusée Agena et une
activité extravéhiculaire (EVA) qui constitue la deuxième sortie américaine
et la troisième en tout, réalisée par Scott. La mission doit durer 75 heures
et le vaisseau doit effectuer 55 orbites. Après le lancement de l'étage-
cible Agena à 15 h 00 UTC, la fusée Titan II GLV transportant Armstrong et
Scott décolle à 16 h 41 UTC. Une fois en orbite, la poursuite de l'étage
Agena par le vaisseau Gemini 8 s'engage."
```

```
liste_termes = re.findall(r"([1-9]\d*|0)", d)
```

Transitions : caractères et groupes de caractères

- a : le caractère a
- [ab] : le caractère a ou le caractère b
- [a-z] : n'importe quel caractère minuscule entre a et z
- [^a] : n'importe quel caractère sauf le caractère a
- * : le caractère espace
- \n : le caractère "passage à la ligne"
- . : n'importe quel caractère
- \. : le caractère "." uniquement
- [1-9] : un chiffre entre 1 et 9
- \w : n'importe quel caractère alphanumérique
- \s : n'importe quel caractère d'espacement
- \d : n'importe quel chiffre

Def : une expression est définie comme une suite de transition. Le mot est accepté lorsque la suite de transitions est respectée

Exemple :

```
r"chal[eu]t"
```

accepte chalet et chalut

```
r"\w\w\w"
```

accepte tous les mots de 3 lettres

Branchements et Récursion

- Les parenthèses permettent :
  - de délimiter une expression (qui peut alors être traitée comme une transition)
    - (artichaud)
  - de définir des branchements :
    - (chien|chat)
- * : le caractère ou l'expression précédente répété entre 0 et n fois
- + : le caractère ou l'expression précédente répété entre 1 et n fois
- ? : le caractère ou l'expression précédente répété entre 0 et 1 fois

exemples

- reconnaître une adresse mail :

```
r'\w[\.\w\-]*\w@\w[\.\w\-]*\.\w\w\w?'
```

3.3 Comparaison/appariement de textes

TODO

3.4 Complétion / Correction

TODO