

## Série 3

# Troisième série : les AEF

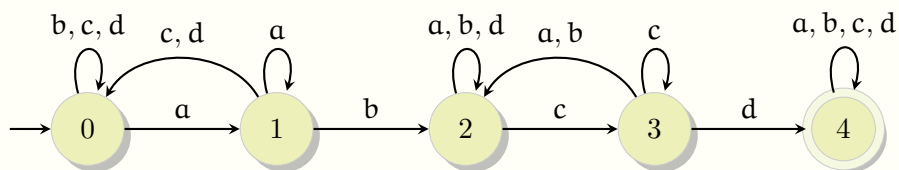
### 1. Éléments nécessaires pour la réalisation des exercices

Afin de réaliser les différentes activités de cette série de TP, il est indispensable de :

- Connaître le fonctionnement des AEF.
- Avoir des notions sur la récursivité
- Suivre le cours/TD du deuxième chapitre

#### Exercice 1

On considère le langage de tous les mots sur  $\{a, b, c, d\}$  ayant le facteur  $ab$  et  $cd$  tel que l'occurrence de  $ab$  précède celle de  $cd$ . On considère alors l'AEF suivant :



#### Activité 1

Ecrire une fonction simulant cet automate. Pour chaque mot analysé, compter le nombre d'étapes nécessaires pour accepter ou rejeter les mots.

#### En-tête

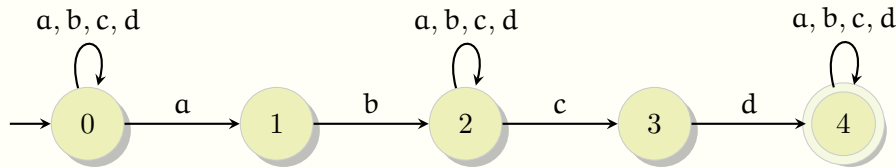
```
def simulate_fsa(word)
```

#### Test

```
simulate_fsa("cdbaabbcd") et simulate_fsa("acdbab")
```

**Exercice 2**

On reprend le langage de l'exercice précédent. On utilise cette fois, l'AEF non-déterministe suivant :

**Activité 2**

Ecrire une fonction simulant cet automate. Pour chaque mot analysé, compter le nombre d'étapes nécessaires pour accepter ou rejeter les mots.

**En-tête**

```
def simulate_fsa(word)
```

**Test**

```
simulate_ndet_fsa("cdbaabbcd") et simulate_ndet_fsa("acdbab")
```

**Exercice 3**

On s'intéresse aux chaînes de caractères contenant 0 et/ou 1 censées représenter des nombres binaires.

**Activité 3**

Ecrire une fonction permettant de calculer le reste de division sur 3 d'un nombre binaire représenté par une chaînes de caractères. Il n'est pas autorisé de transformer la chaîne en un entier ni d'utiliser les opérations d'addition, soustraction, multiplication, division ou modulo.

**En-tête**

```
def remainder_by_3(word)
```

**Test**

```
remainder_by_3("001001111")
remainder_by_3("111011")
remainder_by_3("101010")
```