TMI - 2

TP. n° 6

1. On veut écrire une fonction Maple qui simule un automate fini déterministe .

- Les états sont donnés par des numéros,
- l'alphabet est un sous-ensemble des symboles alphanumériques du clavier,
- la fonction de transition est donnée par une table ; par exemple pour l'automate de l'exercice 1 du TD n° 4 on a :

```
> delta[1,a]:=2;
```

$$\delta_{1,a} := 2$$

```
> delta[1,b]:=3;
```

$$\delta_{1,b} := 3$$

```
> delta[2,a]:=3;
```

$$\delta_{2,a} := 3$$

```
> delta[2,b]:=1;
```

$$\delta_{2,b} := 1$$

```
> delta[3,a]:=2;
```

$$\delta_{3,a} := 2$$

```
> delta[3,b]:=3;
```

$$\delta_{3,b} := 3$$

- les états finaux sont donnés par un ensemble (*set*).

Par exemple un appel possible de la fonction automate est le suivant :

```
> automate(1, {2}, delta, aaaababab);
```

où

- l'entier 1 représente l'état initial,
- l'ensemble {2} représente l'ensemble des états finaux (acceptants),
- *delta* est la table définissant la fonction de transition (définie comme plus haut),
- *aaaababab* est le mot que l'automate doit traiter

Le résultat de l'appel de cette fonction doit être quelque chose du type

```
1 --a--> 2 --a--> 3 --a--> 2 --a--> 3 --b--> 3 --a--> 2 --b--> 1 --a--> 2 --b--> 1
```

Le mot aaaababab est refusé

qui montre toutes les transitions qui ont lieu et à la fin dit si le mot est accepté ou non.
N'oubliez pas de traiter les situations d'erreur où il n'existe pas de transition (parce que c'est la

fonction *delta* qui est comme ça ou parce qu'il y a un caractère qui n'est pas dans l'alphabet. Par exemple:

```
> automate(1, {2}, delta, bac);
1 --b--> 3 --a--> 2
Error, (in automate) Pas de transition depuis l'etat 2 avec la lettre c
```

2. Modifiez le programme précédent pour simuler une machine séquentielle.
L'appel de la fonction sera

```
> seq_machine(1, delta, sigma, MOT);
```

où 1 est l'état initial, *delta* la table des transitions, *MOT* est la chaine en entrée et sigma est la fonction d'affichage. Testez votre code sur une des machines séquentielles du TD 6 ; par exepmle celle qui remplace les groupes de *a* par * et les groupes de *b* par #.