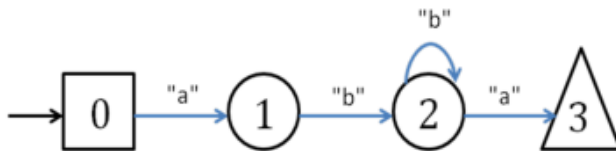


## Objectifs

- Comprendre les principes de base de l'automate à états finis
- Apprendre un des formalismes possibles
- Manipuler quelques cas applicatifs
  - reconnaissance de motifs
  - gestion de flux de navigation

### Exercice 1:



1. Quel est l'alphabet de cet automate ?
2. Quels sont les états de cet automate ? L'état initial et l'état final ?
3. Quel mot cet automate permet-il de reconnaître ?
4. L'automate permet-il de reconnaître un autre mot ? Pourquoi ?

### Réponses

1. "a" et "b"
2. Il a 4 états, 0 (initial), 1, 2 et 3 (final)
3. "abba" par exemple
4. Oui, puisque l'on peut utiliser la transition de l'état 2 vers lui même autant de fois que l'on veut (y compris zéro). On peut former tous les mots de la forme "abb\*a". On a au moins un "b", et le "\*" signifie que l'on peut y ajouter un nombre de "b" supérieur ou égal à zéro. Mais ce n'est pas "ab\*a" puisqu'on a au moins un "b". Au contraire, l'automate ci-dessous, qui n'autorise pas de choix dans les transitions ne reconnaîtra que "abba" :

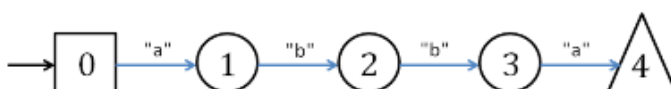
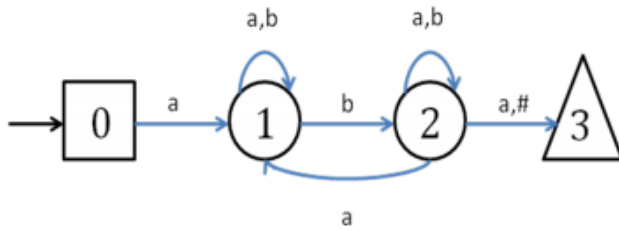


Figure 1: Automate reconnaissant uniquement "abba"

### Exercice 2:



1. Cet automate accepte-t-il le même mot que l'automate de l'exercice 1 ?
2. Quelle est alors la différence ?
3. Pouvez-vous lister les mots acceptés ?

### Réponses

1. Oui, il pourra reconnaître n'importe lequel des mots reconnus par l'automate de l'exercice 1
2. La différence, c'est qu'il en reconnaît d'autres, par exemple je peux boucler sur l'état 1 et former des mots commençant par "aa"
3. On ne peut pas lister les mots puisque le nombre de boucle n'est pas contraint, le nombre de mots reconnaissable est infini. Tous les mots commençant par "a" puis "[ab]\*b" ensuite il est possible de boucler sur l'état 2 avec "[ab]\*" ou de repartir vers l'état 1 avec un "a" et de repartir sur la séquence "[ab]\*b[ab]\*". A l'état 2 on peut aussi choisir la transition vers l'état final avec "a" ou "#".

### Exercice 3:

1. Dessinez un automate ne permettant d'accepter que le mot "amasser".  
L'automate sera purement linéaire puisque l'on ne peut s'autoriser de boucle sur le "s", il en faut 2, pas plus pas moins.
2. Listez les états et l'alphabet de votre automate. 8 états (de 0 l'état initial à 8 l'état final) et un alphabet de 5 éléments : "a", "m", "s", "e", "r"
3. Modifiez l'automate que vous venez de dessiner afin qu'il accepte le mot "amasser", mais aussi le mot "amas". Il suffit de transformer l'état 4 (le premier "s") en état final

### Exercice 4:

Le vocabulaire de l'automate ci-dessous est composé de D, N, V, et P. Avec : D = Déterminant ; N = Nom ; V = Verbe ; P = Ponctuation



Cet automate peut décrire la structure grammaticale d'une phrase. Parmi ces phrases, lesquelles sont des mots du langage décrits par l'automate ?

1. La fille court. **OUI, on a bien une séquence Déterminant, Nom, Verbe, Ponctuation**
2. Le soleil brille. **idem**
3. Les pierres tombent **Il manque la ponctuation**
4. Il fait chaud. **Echoue sur l'item 1**
5. Nous mangeons au restaurant. **Echoue sur l'item 1**
6. Le poisson est rudement bon. **La séquence "rudement bon" (Adverbe Adjectif) n'est pas reconnue par l'automate**

### Exercice 5:

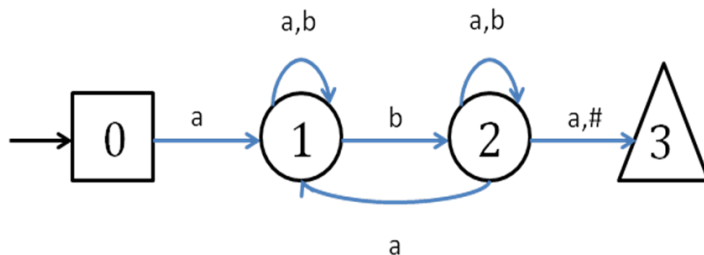


Figure 2: Automate TOTO

Dans le tableau 1a vous trouverez le détail des règles de transition de l'automate TOTO. Par exemple la première ligne que dans l'état 0, je peux suivre la transition "a" qui m'amène dans l'état 1. Comme c'est la seule ligne où 0 est l'état courant, cela signifie que je n'ai qu'une possibilité de transition à partir de cet état. Au contraire, les lignes 2 à 4 de ce même tableau montrent qu'à partir de l'état 1, j'ai plusieurs choix.

État courant	valeur	État de sortie
0	a	1
1	a	1
1	b	2
2	a	2
2	b	2
2	a	1
2	a	3
2	#	3

(a) Règles de transition de TOTO

	a	b	#
→ 0	1		
1	1	1, 2	
2	1, 2, 3	2	3
* 3			

(b) Table de transitions de TOTO

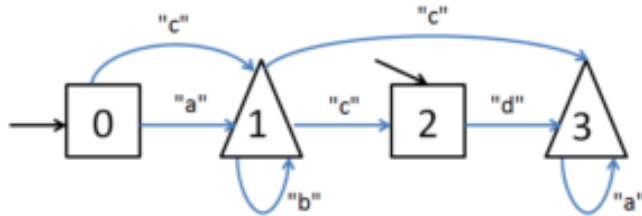
Ces règles de transition peuvent être réduites dans une table de transitions. Les colonnes représentent les symboles de l'alphabet utilisé, et les lignes correspondent aux états. La table de transition est suffisante pour décrire complètement l'automate. C'est une sorte de résumé (cf. tableau 1b)

1. Écrivez la table de transition de l'automate présenté dans l'exercice 2
2. Dessinez l'automate correspondant à la table de transition du tableau 2 :

### Exercice 6:

	a	b
→ 1	1	2
2	1	1, 2

Table 2: Table de transitions



1. Listez les états de cet automate, et indiquez les états finaux et initiaux.
2. Listez l'alphabet.
3. Indiquez la succession d'états permettant d'accepter chacun des mots du tableau ci-après :

Mot	Parcours?	Mot	Parcours?
a		cc	
aacd		d	
aaac		bcd	
acdaa		dd	
ac		b	
abbb		ccaaaaa	

### Exercice 7:

1. Représentez l'automate susceptible de reconnaître une structure de type sujet-verbe-complément (sans entrer dans le détail), en ayant la possibilité d'y intercaler un – et un seul – adverbe. Exemple de mots acceptés : "Il aime beaucoup le travail." et "Ce garçon pense souvent à toi."
2. Représentez l'automate acceptant uniquement les mots "machin", "machine", "machiner", "machinerie". Optimisez l'automate au maximum, en repérant les racines communes de ces mots.

### Exercice 8:

Représentez sous forme d'automate la navigation dans les menus d'une application de jeu, en suivant ces consignes:

- de créer un nouveau personnage, qui amène à l'écran de création de personnage,
- de configurer le jeu, qui amène au panneau de configuration,
- d'aller vers l'éditeur de parties, qui charge cet éditeur,
- de débiter le jeu en chargeant une partie existante.

## Ouverture

Des vidéos d'applications (expérimentales ou pratiques) mettant en œuvre des machines de Turing ou des Automates à Etats Finis.

- La Machine de Turing réalisée Doc. retraçant l'implantation d'un calculateur par des étudiants de l'ENS Lyon avec des Legos :
  - <https://tinyurl.com/turing-realisee>
- Machine de Turing sur Minecraft Utilisation des mécanismes logiques du jeu Minecraft pour implanter un calculateur en 3D. Démonstrations sur quelques calculs simples... avec un mécanisme très complexe:
  - <https://www.youtube.com/watch?v=1X21HQphy6I>