# Théorie des langages
# Introduction aux automates finis

TELECOM Nancy (1A)

2019-2020

## Plan

- Rappels
- Automates finis indéterministes
- Automates finis déterministes
- Expressions régulières et automates (Théorème de Kleene)
  - Des expressions régulières aux automates finis
  - Des automates finis aux expressions régulières

# Rappels

- Opérations sur les langages : union, produit (concaténation), fermeture itérative ...
- L'ensemble $Rat(A^*)$ des langages réguliers (ou rationnels) sur l'alphabet $A$ est défini inductivement par :
  1. la base est l'ensemble $\{\emptyset, \{\varepsilon\}\} \cup \{\{a\}, a \in A\}$
  2. l'ensemble des opérations est $\mathcal{O}p = \{union,\ produit,\ itéré\}$
- L'ensemble $\mathcal{R}_A$ des **expressions régulières** (ou rationnelles) pour l'alphabet $A$ est défini inductivement sur l'alphabet $A \cup \{),\ (,\ \emptyset,\ +,\ *,\ \varepsilon\}$ de la manière suivante :
  1. la base est $\{\emptyset,\ \varepsilon\} \cup \{a, a \in A\}$
  2. si $e_1$ et $e_2$ sont deux expressions régulières, alors $(e_1 + e_2)$, $(e_1 e_2)$, $(e_1)^*$ sont des expressions régulières.
- Résultat : un langage est régulier ssi il est dénoté par une expression régulière.

## Définition

On appelle automate fini (AF) tout quintuplet $\mathcal{A} = (A,\ Q,\ I,\ E,\ T)$ où

- $A$ est un **alphabet**
- $Q$ est un ensemble fini, l'ensemble des **états**
- $I \subset Q$ est l'ensemble des **états initiaux**
- $T \subset Q$ est l'ensemble des **états finaux (terminaux)**
- $E \subset Q \times (A \cup \{\varepsilon\}) \times Q$ est l'ensemble des transitions de $\mathcal{A}$ ($E$ définit une relation appelée **relation de transition** de $\mathcal{A}$)

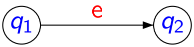
## Remarque

- Une **transition** de $\mathcal{A}$ est un triplet de $E$.
- Une transition est de la forme $(q_1,\ a,\ q_2)$ ou $(q_1, \varepsilon,\ q_2)$ où $q_1$ et $q_2$ sont des états, $a$ est une lettre de $A$ et $\varepsilon$ le mot vide
- On note aussi de telles transitions $q_1 \xrightarrow{a} q_2$ ou $q_1 \xrightarrow{\varepsilon} q_2$.
- $q_1$ et $q_2$ sont respectivement l'origine et l'extrémité de la transition, $a$ et $\varepsilon$ sont les étiquettes des transitions.

# Représentation sagittale d'un automate

Un automate fini $(A,\ Q,\ I,\ E,\ T)$ est représenté par un graphe orienté tel que

- les sommets du graphe sont les états de l'automate, les éléments de $Q$
- toute transition de la forme $(q_1,\ e,\ q_2)$ est représentée par un arc étiqueté par $e \in A \cup \{\varepsilon\}$ et reliant les sommets $q_1$ et $q_2$.

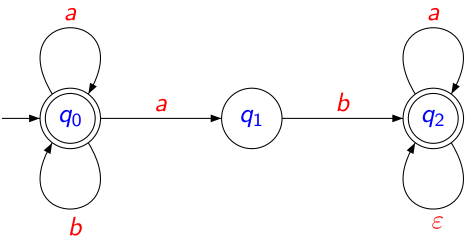



- les états initiaux (éléments de $I$) sont repérés par une pointe de flèche,
- les états finaux (élements de $T$) sont notés par des doubles cercles (ou par des flèches sortantes non reliées à d'autres sommets)

### Exemple

$\mathcal{A} = (\{a,\ b\},\ \{q_0,\ q_1,\ q_2\},\ \{q_0\},\ E,\ \{q_0,\ q_2\})$ où

$E = \{(q_0, a, q_0),\ (q_0, b, q_0),\ (q_0, a, q_1),\ (q_1, b, q_2),\ (q_2, a, q_2),\ (q_2, \varepsilon, q_2)\}$



## Définition

Soit $\mathcal{A} = (A,\ Q,\ I,\ E,\ T)$ un automate fini. On appelle **calcul** dans $\mathcal{A}$ toute suite $c$ de transitions $((q_i, a_i, q_{i+1})_{i\in[1,n-1]}$ tel que l'extrémité d'une transition est l'origine de la suivante. On note un tel calcul de la façon suivante :

$$c = q_1 \xrightarrow{a_1} q_2 \xrightarrow{a_2} q_3 \xrightarrow{a_3} \dots \xrightarrow{a_{n-1}} q_n$$

On dit que :

- $a_1 a_2 a_3 \dots a_{n-1} \in A^*$ est l'étiquette du calcul $c$
- $q_1 \in Q$ est l'origine de $c$
- $q_n \in Q$ est l'extrémité de $c$

## Exemple

Le calcul défini par :

$$c = \big((1, a, 1), (1, b, 2), (2, b, 3), (3, a, 2)\big) = 1 \xrightarrow{a} 1 \xrightarrow{b} 2 \xrightarrow{b} 3 \xrightarrow{a} 2$$

a pour origine l'état 1, pour extrémité l'état 2 et pour étiquette le mot *abba*.

## Remarque

On s'autorise désormais à écrire $q \xrightarrow{\alpha} q'$ pour désigner un calcul d'origine $q$, d'étiquette $\alpha \in A^*$ et d'extrémité $q'$ sans expliciter les états intermédiaires.

## Définition

On dit qu'un calcul dans un automate $\mathcal{A} = (A, Q, I, E, T)$ est **réussi** lorsque son origine appartient à $I$ (*i.e.* est un état initial) et son extrémité à $T$ (*i.e.* est un état final) :

$$c \text{ est un calcul réussi} \quad \Leftrightarrow \quad c = p \xrightarrow{\alpha} q,\ p \in I, q \in T,\ \alpha \in A^*$$

### Définition (état accessible)

Soit $\mathcal{A} = (A, Q, I, E, T)$ un automate fini et $q \in Q$. On dit que $q$ est un état **accessible** si et seulement si il existe un mot $\alpha \in A^*$ et un état initial $q_0 \in I$ tel que $q_0 \xrightarrow{\alpha} q$, autrement dit, s'il existe un calcul dont l'origine est un état intial et dont l'extrémité est $q$.

L'ensemble des états accessibles de $\mathcal{A}$ est donc :

$$\left\{q \in Q,\ (\exists \alpha \in A^*)\ (\exists q_0 \in I)\ q_0 \xrightarrow{\alpha} q\right\}$$

Un état qui n'est pas accessible est dit **inaccessible**.

## Définition

On appelle **automate fini déterministe** tout quintuplet $\mathcal{A} = (A, Q, q_0, \delta, T)$ tel que :

- $A$ est un alphabet
- $Q$ est un ensemble fini, l'ensemble des états
- $q_0 \in Q$ est l'unique état initial
- $\delta : Q \times A \to Q$ est la fonction transition
- $T \subset Q$ est l'ensemble des états terminaux

## Remarque

Un automate fini déterministe est un cas particulier d'un automate fini indéterministe avec les contraintes suivantes :

- $I$, l'ensemble des états initiaux, est un singleton
- $E$, l'ensemble des transitions, est défini par la fonction $\delta$ : c.-à-d. pour tout état $q \in Q$ et toute lettre $a \in A$ il existe au plus un état $q' \in Q$ tel que $q \xrightarrow{a} q'$
- d'après la définition de la fonction $\delta$, il n'y a pas de transition étiquetée par $\varepsilon$

### Remarque

Soit $\mathcal{A} = (A, Q, I, E, T)$ un automate, si $\mathcal{A}$ comporte au moins une des caractéristiques suivantes :

- *i* plusieurs états initiaux
- *ii* au moins une transition étiquetée par $\varepsilon$
- *iii* un état qui est l'origine de plusieurs transitions différentes de même étiquette, c'est-à-dire qu'il existe au moins deux transitions $(q,\ a,\ q')$ et $(q,\ a,\ q'')$ avec $q' \neq q''$.

alors $\mathcal{A}$ est indéterministe, sinon, il est déterministe.

# Prolongement de la fonction de transition $\delta$

## Définition

Soit $(Q,\ A,\ \delta,\ s_0,\ F)$ un automate fini déterministe, la fonction de transition $\delta\ :\ Q \times A \rightarrow Q$ se prolonge en une application $\delta^*\ :\ Q \times A^* \rightarrow Q$ définie par :

- $\delta^*(q,\ \varepsilon) = q$ pour tout $q$ de $Q$
- $\delta^*(q,\ \alpha.a) = \delta(\delta^*(q,\ \alpha),\ a)$ pour tout $q$ de $Q$, pour tout $\alpha$ de $A^*$ et tout $a$ de $A$

## Remarque et propriété

- $\delta^*$ est définie par récurrence sur la longueur du mot, (définition pour la base $\varepsilon$ et les mots $\alpha a$)
- $\delta^*$ est un prolongement de $\delta$, (c.-à-d. $\delta^*(q, a) = \delta(q, a)$ pour tout état $q$ et toute lettre $a$)
- ce prolongement vérifie : $\delta^*(q,\ \alpha.\beta) = \delta^*(\delta^*(q,\ \alpha),\ \beta)$ pour tout état $q$ de $Q$ et tout mot $\alpha$ et $\beta$ de $A^*$ (démonstration à effectuer par récurrence sur la longueur de $\beta$)

## Définition

Soit $\mathcal{A} = (A, Q, I, E, T)$ un automate fini. On dit qu'un mot $\alpha \in A^*$ est **reconnu** (ou **accepté**) par $\mathcal{A}$ s'il existe un calcul réussi d'étiquette $\alpha$ :

$$(\exists (q_0, \ldots, q_n) \in Q^{n+1})\ (\exists (a_0, \ldots, a_{n-1}) \in (A \cup \{\varepsilon\})^n)$$

$$q_0 \in I \text{ et } q_n \in T \text{ et } q_0 \xrightarrow{a_0} q_1 \ldots q_{n-1} \xrightarrow{a_{n-1}} q_n$$

où $\alpha = a_0 \ldots a_{n-1}$. Ce qui s'écrit, sous forme plus condensée :

$$(\exists q_0 \in I)\ (\exists q_n \in T)\ q_0 \xrightarrow{\alpha} q_n$$

## Définition

Soit $\mathcal{A} = (A, Q, I, E, T)$, on appelle **langage reconnu** par $\mathcal{A}$ que l'on note $\mathcal{L}(\mathcal{A})$, l'ensemble des mots reconnus par $\mathcal{A}$ :

$$\mathcal{L}(\mathcal{A}) = \left\{\alpha \in A^*,\ (\exists q \in I)\ (\exists q' \in T)\ q \xrightarrow{\alpha} q'\right\}$$

## Définition

On dit qu'un langage est **reconnaissable par un AF** s'il existe un automate fini qui le reconnaît. On note $Rec(A^*)$ l'ensemble des langages sur l'aphabet $A$ reconnaissables par un automate fini.

## Remarque

Si $\mathcal{A} = (A,\ Q,\ q_0,\ \ \delta,\ T)$ un automate fini déterministe :

- Un mot $\alpha$ est reconnu par $\mathcal{A}$ ssi $\delta^*(q_0,\ \alpha) \in T$
- $L(\mathcal{A}) = \{\alpha\ ;\ \delta^*(q_0,\ \alpha) \in T\ \}$

## Remarque

- Etant donné un mot et un automate **déterministe**, il est facile de vérifier si le mot est reconnu par l'automate en utilisant $\delta^*$ (de façon pratique, on peut aussi le faire en utilisant la représentation sagittale d'un automate)
- Si l'automate est **indéterministe** il faut simuler toutes les exécutions en tenant compte des différents choix possibles.

### Théorème

Soit $A$ un alphabet, $Rat(A^*)$ l'ensemble des langages rationnels (réguliers) sur $A$ et $Rec(A^*)$ l'ensemble des langages sur $A$ reconnaissables par un automate fini. Le **théorème de Kleene** établit que :

$$Rat(A^*) = Rec(A^*)$$

Autrement dit, les langages rationnels sont exactement ceux reconnus par les automates finis.

### Remarque

Ce théorème nous permet de dire que pour toute expression régulière dénotant un langage $L$, on peut construire un automate reconnaissant $L$ et, réciproquement, à tout automate $\mathcal{A}$ on peut associer une expression régulière dénotant $\mathcal{L}(\mathcal{A})$.

### Lemme 1

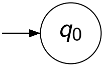
Si un langage est dénoté par une expression régulière (i.e. est régulier) il est reconnu par un automate fini (non déterministe).

### Démonstration

La démonstration utilise la définition inductive des expressions régulières. On montre que pour chaque expression régulière de base ($\emptyset$, $\varepsilon$, $a \in A$) il existe un automate reconnaissant le langage dénoté par cette expression régulière.
On montre que pour des expressions régulières composées $(e_1 + e_2)$, $(e_1 e_2)$ et $(e_1)^*$ il est possible d'obtenir un automate reconnaissant le langage décrit par ces expressions à partir d'automates reconnaissant les langages dénotés par $e_1$ et $e_2$.

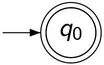
# Démonstration

- à l'expression rationnelle vide, $\varnothing$, on associe l'automate $\mathcal{A} = (A, Q, I, E, T) = (A,\ \{q_0\},\ \{q_0\},\ \emptyset,\ \emptyset)$.



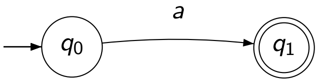

  Cet automate ne reconnaît aucun mot donc $\mathcal{L}(\mathcal{A}) = \emptyset = \mathcal{L}_{\mathcal{R}}(\emptyset)$

- à l'expression rationnelle $\varepsilon$, on associe l'automate $\mathcal{A} = (A,\ \{q_0\},\ \{q_0\},\ \emptyset,\ \{q_0\})$.



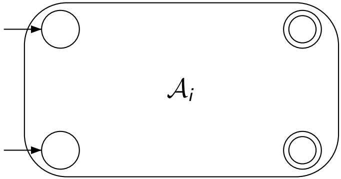

  Cet automate reconnaît le mot vide $\mathcal{L}(\mathcal{A}) = \{\varepsilon\} = \mathcal{L}_{\mathcal{R}}(\varepsilon)$

à tout $a \in A$ on associe l'automate
$\mathcal{A} = \big(A,\ \{q_0, q_1\},\ \{q_0\}, \{(q_0, a, q_1)\}, \{q_1\}\big)$ :

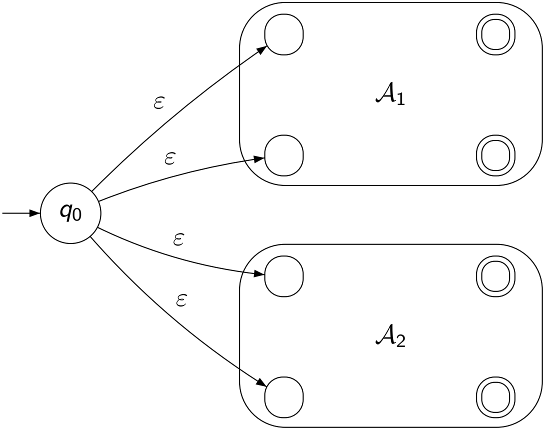



Cet automate reconnaît le mot $a$, donc $\mathcal{L}(\mathcal{A}) = \{a\} = \mathcal{L}_{\mathcal{R}}(a)$

**Hypothèse de récurrence** : Soient $\mathcal{A}_1 = (A,\ Q_1,\ I_1,\ E_1,\ T_1)$ et $\mathcal{A}_2 = (A,\ Q_2,\ I_2,\ E_2,\ T_2)$ deux automates reconnaissant respectivement les langages dénotés par les expressions $e_1$ et $e_2$. L'automate $\mathcal{A}_i$ est représenté par le schéma suivant :

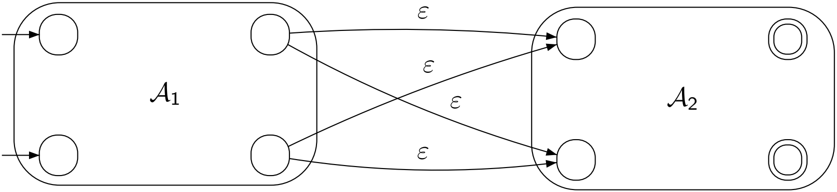


Soit $\mathcal{A}$ l'automate reconnaissant $\mathcal{L}_{\mathcal{R}}\big((e_1 + e_2)\big)$ :




On crée un nouvel état initial $q_0$ et un ensemble $E'$ de nouvelles transitions telles que, $\forall q \in I_1 \cup I_2,\ q_0 \xrightarrow{\varepsilon} q$. On a $\mathcal{A} = (A,\ Q_1 \cup Q_2 \cup \{q_0\},\ \{q_0\},\ E_1 \cup E_2 \cup E',\ T_1 \cup T_2)$.

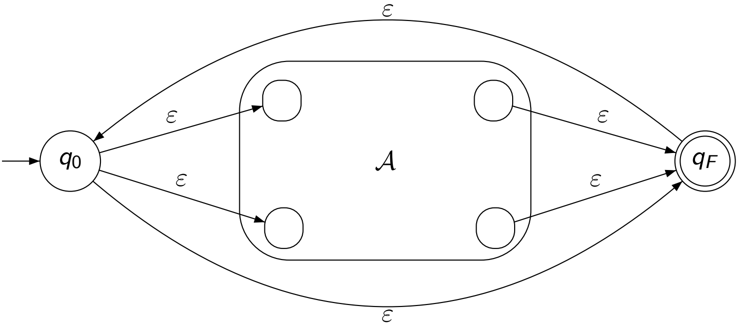
Soit $\mathcal{A}$ l'automate reconnaissant $\mathcal{L}_{\mathcal{R}}\big((e_1 e_2)\big)$ :




On crée un ensemble $E'$ de $\varepsilon$-transitions reliant tous les états terminaux de $\mathcal{A}_1$ aux états initiaux de $\mathcal{A}_2$. On a alors
$\mathcal{A} = (A,\ Q_1 \cup Q_2,\ I_1,\ E_1 \cup E_2 \cup E',\ T_2)$.

# Démonstration $((e)^*)$

Soit $\mathcal{A} = (A, Q, I, E, T)$ l'automate reconnaissant le langage dénoté par l'expression rationnelle $e$. L'automate $\mathcal{A}' = (A,\ Q',\ I',\ E',\ T')$ reconnaissant $\mathcal{L}_{\mathcal{R}}\big((e)^*\big)$ est tel que :




$Q' = Q \cup \{q_0, q_F\},\ \ I' = \{q_0\},\ \ T' = \{q_F\}$ et
$E' = E \cup \{q_0 \xrightarrow{\varepsilon} q, q \in I\} \cup \{q \xrightarrow{\varepsilon} q_F, q \in F\} \cup \{q_0 \xrightarrow{\varepsilon} q_F, q_F \xrightarrow{\varepsilon} q_0\}$.
L'automate reconnaissant $\mathcal{L}_{\mathcal{R}}\big((e)^+\big)$ s'obtient, quant à lui, de la même façon que le précédent en retirant la transition $q_0 \xrightarrow{\varepsilon} q_F$ (dont le but est uniquement de faire reconnaître le mot vide par l'automate).

## Définition

Soit $\mathcal{A} = (A,\ Q,\ I,\ E,\ T)$ un automate fini, soient $p, q \in Q$

- on note $E_{p,q}$, l'ensemble des étiquettes des transitions d'origine $p$ et d'extrémité $q$
- on pose $L_p = \{\alpha\ ;\ \alpha \in A^*\ \mathit{et}\ (\exists r \in T)\ p \xrightarrow{\alpha} r\}$

## Remarques

- $L_p$ est l'ensemble des mots qui sont les étiquettes des calculs qui vont de $p$ à un état final de $\mathcal{A}$
- à chaque état $p \in Q$ on peut associer un langage $L_p$
- on a évidemment $\mathcal{L}(\mathcal{A}) = \sum_{p \in I} L_p$

# Système d'équations

Pour montrer que $\mathcal{L}(\mathcal{A})$ est rationnel il suffit de montrer que chacun des $L_p$ est rationnel.

$$L_p = \{\alpha \ ; \ \alpha \in A^* \ et \ (\exists r \in T) \ p \xrightarrow{\alpha} r\}$$

Un mot $\alpha$ appartient à $L_p$ :

- soit $\alpha = \varepsilon$ dans ce cas $p \in T$ ($p$ est un état final)
- soit $\alpha = a\beta$ avec $a$ l'étiquette d'une transition de $\mathcal{A}$ qui va de $p$ à $q$ et $\beta$ appartient à $L_q$

On peut donc écrire :

$$(\forall p \in Q) \ \ L_p = \sum_{q \in Q} E_{p,q} L_q + \delta_{p,T} \quad (I)$$

où $\delta_{p,T} = \varepsilon$ si $p \in T$ et $\emptyset$ sinon

($I$) définit un système de $n$ équations linéaires à $n$ inconnues avec $n = card(Q)$, les inconnues étant les $L_p$, les coefficients dans $\mathcal{P}(A)$.

On résout ce système par application du "lemme d'Arden".

## Lemme d'Arden

Soit $A$ un vocabulaire et $L_1$ et $L_2$ deux langages sur $A$, on considère l'équation $X = L_1.X + L_2$ (1) où $X$ l'inconnue est un langage sur $A$.

- Si $\varepsilon \notin L_1$ l'équation (1) admet une solution unique $L_1^*.L_2$.
- Si $\varepsilon \in L_1$, l'ensemble des solutions de l'équation (1) est $\{L_1^*.(L_2 + L)\}$ où $L$ est un langage quelconque

## Théorème

Soit le système suivant de $n$ ($n > 0$) équations :

$$\left\{ \; L_i = \sum_{j=1}^{n} E_{i,j}.L_j + F_i \; \right\}_{i \in [1,n]}$$

tels que les langages $E_{i,j}$ $(1 \leq i, j \leq n)$ ne contiennent pas $\varepsilon$ et les langages $L_i$ sont les inconnues. Ce système admet une unique solution (et si les langages $E_{i,j}$ et $F_i$ sont réguliers alors les langages $L_i$ le sont aussi).

La démonstration de ce théorème peut se faire par récurrence sur l'entier $n$, en utilisant le lemme d'Arden.

# Méthode de construction d'une expression régulière à partir d'un automate fini

Soit $\mathcal{A} = (A,\ Q,\ I,\ E,\ T)$ un automate fini sans transition sur le mot vide :

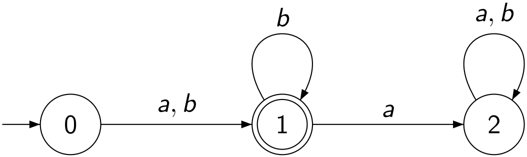
- Construction du système d'équations sous la forme :

$$\left\{\ L_i = \sum_{j=0}^{n} E_{i,j}.L_j + F_i\ \right\}_{i\in[0,n]}$$

  Les états sont numérotés de 0 à $n$. Pour établir ce système on utilise les transitions en partant de chaque état de $\mathcal{A}$

- Résolution du système par substitution, en utilisant le lemme d'Arden sur certaines équations. Le but de la résolution est de donner une expression régulière correspondant à la réunion des langages $L_i$ où $i$ correspond à un état initial de l'automate (on obtient ainsi le langage reconnu par l'automate donné).

Soit $\mathcal{A}$ l'automate suivant :




Ecriture du sytème :

$$\begin{cases} L_0 = & aL_1 + bL_1 = (a+b)L_1 \\ L_1 = & bL_1 + aL_2 + \varepsilon \ \ (car\ 1 \in T) \\ L_2 = & (a+b)L_2 \end{cases}$$

On applique le lemme d'Arden à la 3ème équation

$$\begin{cases} L_0 = & (a+b)L_1 \\ L_1 = & bL_1 + aL_2 + \varepsilon \\ L_2 = & (a+b)^*\emptyset = \emptyset \end{cases}$$

On reporte $L_2 = \emptyset$ dans la deuxième équation :

$$\begin{cases} L_0 = & (a+b)L_1 \\ L_1 = & bL_1 + \varepsilon \\ L_2 = & \emptyset \end{cases}$$

On applique le lemme d'Arden à la deuxième équation :

$$\begin{cases} L_0 = & (a+b)L_1 \\ L_1 = & b^*\varepsilon = b^* \\ L_2 = & \emptyset \end{cases}$$

On reporte $L_1$ dans la première équation en utilisant la deuxième équation :

$$\begin{cases} L_0 = & (a+b)b^* \\ L_1 = & b^* \\ L_2 = & \emptyset \end{cases}$$

On obtient donc $\mathcal{L}(\mathcal{A}) = L_0 = (a+b)b^*$

# Conclusion

- Ce qu'il faut retenir :
  - Automate fini indéterministe (calcul, langage reconnu par un automate)
  - Automate fini déterministe
  - Théorème de Kleene :
    langage régulier (ou rationnel) $\equiv$ langage reconnu par un automate fini)
    - Expression régulière $\longrightarrow$ Automate fini
    - Automate fini $\longrightarrow$ Expression régulière
- Nombreuses applications (dans divers domaines) des automates finis.
- Prochaines notions abordées en TD :
  - Déterminisation des automates
  - Minimisation des automates