Automates, Langages et Logique 3. Automates finis non déterministes

L2, Université d'Orléans — S1 2025/2026

Nicolas Ollinger

Rappels

Un automate fini déterministe $\mathcal{A} = (Q, A, \delta, q_0, F)$ reconnait le langage $L(\mathcal{A}) \in \operatorname{Rec} A^*$ des mots étiquettant des chemins acceptants de l'automate.

Un langage est **reconnaissable** s'il est reconnu par un AFD. Dans ce cas, il est reconnu par un AFD **complet** avec au plus deux états puits.

Deux automates sont **équivalents** s'ils reconnaissent le même langage.

Automate fini non déterministe (AFN)

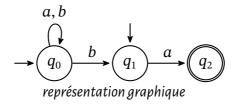
Définition Un automate fini non déterministe (noté AFN dans la suite) \mathcal{A} est un quintuplet (Q, A, T, I, F) avec • *O* l'ensemble fini des états : • A l'alphabet d'entrée; • $T \subseteq Q \times A \times Q$ la relation de transition; • $I \subseteq Q$ l'ensemble des états initiaux; • $F \subseteq Q$ l'ensemble des **états acceptants** de l'automate.

Automate fini non déterministe (AFN)

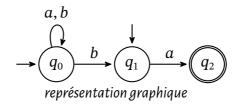
Définition Un automate fini non déterministe (noté AFN dans la suite) \mathcal{A} est un quintuplet (Q, A, T, I, F) avec • *O* l'ensemble fini des états : • A l'alphabet d'entrée; • $T \subseteq Q \times A \times Q$ la relation de transition; • $I \subseteq Q$ l'ensemble des états initiaux; • $F \subseteq Q$ l'ensemble des **états acceptants** de l'automate.

On relâche les contraintes de **déterminisme**. Qu'est-ce que cela change?

3 modes de représentation



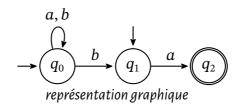
3 modes de représentation



	а	b
$\rightarrow q_0$	q_0	q_0, q_1
$\rightarrow q_1$	q_2	Ø
$q_2 *$	Ø	Ø

représentation tabulaire

3 modes de représentation



	а	b
$\rightarrow q_0$	q_0	q_0, q_1
$\rightarrow q_1$	q_2	Ø
$q_2 *$	Ø	Ø

tabulaire — états en lignes

	$\downarrow q_0$	$\downarrow q_1$	$q_2 *$
a	q_0	q_2	Ø
b	q_0, q_1	Ø	Ø

tabulaire — états en colonnes

Fermeture transitive réflexive

La relation de transition est étendue aux mots par fermeture.

La **relation de transition étendue** applique successivement la relation de transition aux différentes lettres qui composent le mot.

Fermeture transitive réflexive

La relation de transition est étendue aux mots par fermeture.

La **relation de transition étendue** applique successivement la relation de transition aux différentes lettres qui composent le mot.

 T^* est la plus petite relation qui satisfait pour tout **état** $q \in Q$, **lettre** $x \in A$ et **mot** $u \in A^*$: $(q, \varepsilon, q) \in T^* \qquad (q, x, q') \in T^* \qquad \operatorname{si}(q, x, q') \in T$ $(q, u \cdot x, q') \in T^* \qquad \operatorname{si}\left\{ (q, u, q'') \in T^* \right\} \quad \text{pour un certain } q'' \in Q$

Chemin dans un automate

La **relation de transition** T décrit les transitions $q \stackrel{a}{\rightarrow} q'$ possibles dans l'automate.

Chemin dans un automate

La **relation de transition** T décrit les transitions $q \stackrel{u}{\rightarrow} q'$ possibles dans l'automate.

Définition Un chemin dans un AFN (Q, A, T, I, F) est une suite finie de transitions successives $p_0 \xrightarrow{x_0} p_1 \xrightarrow{x_1} \cdots \xrightarrow{x_{n-1}} p_n$ où p_0 est l'état de départ, p_n est l'état d'arrivée et $(p_i, x_i, p_{i+1}) \in T$ pour tout i < n. Le mot $x_0 \cdots x_{n-1}$ est l'étiquette du chemin.

Chemin dans un automate

La **relation de transition** T décrit les transitions $q \stackrel{a}{\rightarrow} q'$ possibles dans l'automate.

Définition Un chemin dans un AFN (Q, A, T, I, F) est une suite finie de transitions successives $p_0 \xrightarrow{x_0} p_1 \xrightarrow{x_1} \cdots \xrightarrow{x_{n-1}} p_n$ où p_0 est l'état de départ, p_n est l'état d'arrivée et $(p_i, x_i, p_{i+1}) \in T$ pour tout i < n. Le mot $x_0 \cdots x_{n-1}$ est l'étiquette du chemin.

En utilisant la **relation de transition étendue** on s'autorise une notation plus concise $p_0 \xrightarrow[]{x_0x_1\cdots x_{n-1}} p_n$ lorsque $(p_0, x_0x_1\cdots x_{n-1}, p_n) \in T^*$.

Définition Un chemin est acceptant pour un AFN lorsque l'état de départ est un **état initial** et l'état d'arrivée un **état acceptant**.

$$I \ni q_0 \stackrel{u}{\to} q_f \in F \qquad (q_0, u, q_f) \in T^*$$

Définition Un chemin est acceptant pour un AFN lorsque l'état de départ est un **état initial** et l'état d'arrivée un **état acceptant**.

$$I \ni q_0 \xrightarrow{u} q_f \in F \qquad (q_0, u, q_f) \in T^*$$

Définition Un mot u est **accepté** par un automate $\mathcal A$ s'il est l'étiquette d'un **chemin acceptant** de $\mathcal A$.

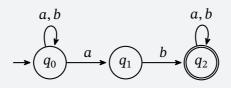
Définition Un chemin est acceptant pour un AFN lorsque l'état de départ est un **état initial** et l'état d'arrivée un **état acceptant**.

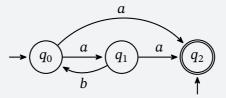
$$I \ni q_0 \xrightarrow{u} q_f \in F \qquad (q_0, u, q_f) \in T^*$$

Définition Un mot u est **accepté** par un automate \mathcal{A} s'il est l'étiquette d'un **chemin acceptant** de \mathcal{A} .

Définition Le langage reconnu par un automate $\mathcal A$ est l'ensemble $L(\mathcal A)$ des mots acceptés par $\mathcal A$.

Exercice Décrire les **langages reconnus** par les automates finis non déterministes suivants.





Comment définir une fonction de transition sur un AFN?

Comment définir une fonction de transition sur un AFN?

Définition La fonction de transition $\delta: Q \times A \to \mathcal{P}(Q)$ d'un AFN (Q,A,T,I,F) est définie pour tout état $q \in Q$ et lettre $x \in A$ par $\delta(q,x) = \{q' \in Q | (q,x,q') \in T\}$.

Comment définir une fonction de transition sur un AFN?

Définition La fonction de transition $\delta: Q \times A \to \mathcal{P}(Q)$ d'un AFN (Q,A,T,I,F) est définie pour tout état $q \in Q$ et lettre $x \in A$ par $\delta(q,x) = \{q' \in Q | (q,x,q') \in T\}$.

La fonction de transition étendue capture les états atteints.

Comment définir une fonction de transition sur un AFN?

Définition La fonction de transition $\delta: Q \times A \to \mathcal{P}(Q)$ d'un AFN (Q,A,T,I,F) est définie pour tout état $q \in Q$ et lettre $x \in A$ par $\delta(q,x) = \{q' \in Q | (q,x,q') \in T\}$.

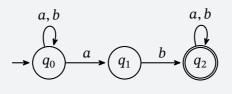
La fonction de transition étendue capture les états atteints.

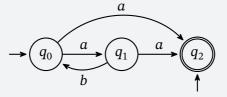
Définition La fonction de transition étendue $\delta^*: \mathcal{P}(Q) \times A^* \to \mathcal{P}(Q)$ d'un AFN (Q,A,T,I,F) est définie récursivement pour tout ensemble d'états $E \subseteq Q$, mot $u \in A^*$ et lettre $x \in A$ par

$$\delta^*(E,\varepsilon) = E$$

$$\delta^*(E, u \cdot x) = \bigcup_{q \in \delta^*(E,u)} \delta(q, x)$$

Exercice Calculer $\delta^*(E, x)$ pour tout $E \subseteq Q$ et $x \in \{a, b\}$ pour chacun des automates suivants.





Les automates non déterministes n'apportent pas d'expressivité...

Les automates non déterministes n'apportent pas d'expressivité...

Proposition Tout AFN est équivalent à un AFD.

Les automates **non déterministes** n'apportent pas d'**expressivité**...

Proposition Tout AFN est **équivalent** à un AFD.

...ils apportent parfois de la concision!

Les automates non déterministes n'apportent pas d'expressivité...

Proposition Tout AFN est **équivalent** à un AFD.

...ils apportent parfois de la concision!

Proposition La familles des **langages reconnaissables** est stable par miroir, *i.e.* si $L \in \operatorname{Rec} A^*$ alors $L^R = \{u^R | u \in L\} \in \operatorname{Rec} A^*$.

Exercice Soit L_n le langage des mots sur l'alphabet $\{a,b\}$ qui contiennent un a en $n^{\rm e}$ position en partant de la fin du mot.

- (a) Donner une expression régulière pour L_n ;
- **(b)** Proposer un AFN à O(n) états pour L_n ;
- (c) Combien d'états pour un AFD reconnaissant L_n ?

Transitions enrichies

Que se passe-t-il si on **étiquette** les **transitions** par des **mots** $u \in A^*$ plutôt que par des **lettres** $x \in A$?

Transitions enrichies

Que se passe-t-il si on **étiquette** les **transitions** par des **mots** $u \in A^*$ plutôt que par des **lettres** $x \in A$?

Remarque Une transition $q \xrightarrow{u_0 \cdots u_n} q'$ étiquetée par un mot non vide peut être transformée par ajout de nouveaux états en une suite de transitions $q = q_0 \xrightarrow{u_0} q_1 \xrightarrow{u_1} q_2 \cdots \xrightarrow{u_n} q_{n+1} = q'$.

Transitions enrichies

Que se passe-t-il si on **étiquette** les **transitions** par des **mots** $u \in A^*$ plutôt que par des **lettres** $x \in A$?

Remarque Une transition $q \xrightarrow{u_0 \cdots u_n} q'$ étiquetée par un mot non vide peut être transformée par ajout de nouveaux états en une suite de transitions $q = q_0 \xrightarrow{u_0} q_1 \xrightarrow{u_1} q_2 \cdots \xrightarrow{u_n} q_{n+1} = q'$.

Mais...et le **mot vide** alors?

Transitions spontanées

Une ε -transition est une transition spontanée $q \stackrel{\varepsilon}{\to} q'$.

Transitions spontanées

Une ε -transition est une transition spontanée $q \stackrel{\varepsilon}{\to} q'$.

Définition Un automate fini non déterministe avec transitions spontanées (noté ε -AFN dans la suite) $\mathcal A$ est un quintuplet (Q,A,T,I,F) avec

- *Q* l'ensemble fini des **états**;
- *A* l'alphabet d'entrée;
- $T \subseteq Q \times (A \cup \{\epsilon\}) \times Q$ la relation de transition;
- $I \subseteq Q$ l'ensemble des **états initiaux**;
- $F \subseteq Q$ l'ensemble des **états acceptants** de l'automate.

Fermeture transitive réflexive

La **relation de transition étendue** applique successivement la relation de transition aux différentes lettres qui composent le mot.

 T^* est la plus petite relation qui satisfait pour tout **état** $q \in Q$, **mot** $x \in A \cup \{\varepsilon\}$ et **mot** $u \in A^*$: $(q, \varepsilon, q) \in T^*$ $(q, x, q') \in T^* \quad \text{si } (q, x, q') \in T$ $(q, u \cdot x, q') \in T^* \quad \text{si } \begin{cases} (q, u, q'') \in T^* \\ (q'', x, q') \in T \end{cases}$ pour un certain $q'' \in Q$

Chemin dans un ε -AFN

La définition de chemin reste la même.

Chemin dans un ε -AFN

La définition de chemin reste la même.

Définition Un chemin dans un ε -AFN (Q,A,T,I,F) est une suite finie de transitions successives $p_0 \xrightarrow{x_0} p_1 \xrightarrow{x_1} \cdots \xrightarrow{x_{n-1}} p_n$ où p_0 est l'**état de départ**, p_n est l'**état d'arrivée** et $(p_i,x_i,p_{i+1}) \in T$ pour tout i < n. Le mot $x_0 \cdots x_{n-1}$ est l'**étiquette** du chemin.

Chemin dans un ε -AFN

La définition de chemin reste la même.

Définition Un chemin dans un ε -AFN (Q,A,T,I,F) est une suite finie de transitions successives $p_0 \xrightarrow{x_0} p_1 \xrightarrow{x_1} \cdots \xrightarrow{x_{n-1}} p_n$ où p_0 est l'**état de départ**, p_n est l'**état d'arrivée** et $(p_i,x_i,p_{i+1}) \in T$ pour tout i < n. Le mot $x_0 \cdots x_{n-1}$ est l'**étiquette** du chemin.

En utilisant la **relation de transition étendue** on s'autorise une notation plus concise $p_0 \xrightarrow[x_0x_1\cdots x_{n-1}]{x_0x_1\cdots x_{n-1}} p_n$ lorsque $(p_0,x_0x_1\cdots x_{n-1},p_n)\in T^*$.

Définition Un chemin est acceptant pour un AFN lorsque l'état de départ est un **état initial** et l'état d'arrivée un **état acceptant**.

$$I \ni q_0 \stackrel{u}{\to} q_f \in F \qquad (q_0, u, q_f) \in T^*$$

Définition Un chemin est acceptant pour un AFN lorsque l'état de départ est un **état initial** et l'état d'arrivée un **état acceptant**.

$$I \ni q_0 \xrightarrow{u} q_f \in F \qquad (q_0, u, q_f) \in T^*$$

Définition Un mot u est **accepté** par un automate \mathcal{A} s'il est l'étiquette d'un **chemin acceptant** de \mathcal{A} .

Définition Un chemin est acceptant pour un AFN lorsque l'état de départ est un **état initial** et l'état d'arrivée un **état acceptant**.

$$I \ni q_0 \xrightarrow{u} q_f \in F \qquad (q_0, u, q_f) \in T^*$$

Définition Un mot u est **accepté** par un automate \mathcal{A} s'il est l'étiquette d'un **chemin acceptant** de \mathcal{A} .

Définition Le langage reconnu par un automate $\mathcal A$ est l'ensemble $L(\mathcal A)$ des mots acceptés par $\mathcal A$.

Exercice Soit $L \in \text{Rec } A^*$, montrer que les langages des suffixes et facteurs de L sont reconnus par des ε -AFN.

Idée Supposer que L est reconnu par un AFD émondé.

Élimination des ε -transitions

La notion de langage reconnaissable est très robuste.

Élimination des ε -transitions

La notion de langage reconnaissable est très robuste.

Proposition Tout ε -AFN est **équivalent** à un AFN (et donc à un AFD).

Idée procéder à la fermeture (avant ou arrière) des transitions par la clôture transitive des transitions spontanées.

Élimination des ε -transitions

La notion de langage reconnaissable est très robuste.

Proposition Tout ε -AFN est **équivalent** à un AFN (et donc à un AFD).

Idée procéder à la fermeture (avant ou arrière) des transitions par la clôture transitive des transitions spontanées.

Proposition Tout AFN étiqueté par des mots est équivalent à un AFN.

Morphismes sur les mots

Le langage A^* muni de la concaténation et du mot vide ε est un monoïde.

Définition Soient (E, \times, e) et (F, \cdot, f) deux monoïdes. Un morphisme de (E, \times, e) dans (F, \cdot, f) est une application $\varphi : E \to F$ telle que : (i) $\varphi(e) = f$;

(ii) $\varphi(x \times y) = \varphi(x) \cdot \varphi(y)$ pour tous $x, y \in E$.

Remarque Un morphisme sur les mots est donc une application $\varphi: A^* \to B^*$ telle que $\varphi(\varepsilon) = \varepsilon$ et $\varphi(uv) = \varphi(u)\varphi(v)$ pour tous mots $u, v \in A^*$. Il est donc complétement défini par sa restriction à A.

Stabilité

Grâce aux automates non-déterministes étiquetés par des mots, on obtient facilement de nouvelles propriétés de clôture.

Proposition La familles des langages reconnaissables est stable par :		
(a) concaténation;	$L_1, L_2 \in \operatorname{Rec} A^* \Rightarrow L_1 \cdot L_2 \in \operatorname{Rec} A^*$	
(b) étoile de Kleene;	$L \in \operatorname{Rec} A^* \Rightarrow L^* \in \operatorname{Rec} A^*$	
(c) image par un morphisme;	$L \in \operatorname{Rec} A^* \Rightarrow \varphi(L) \in \operatorname{Rec} A^*$	
(d) image inverse.	$L \in \operatorname{Rec} A^* \Rightarrow \varphi^{-1}(L) \in \operatorname{Rec} A^*$	

Langages rationnels

Définition La famille $\operatorname{Rat} A^*$ des **langages rationnels** sur l'alphabet A est la plus petite famille de langages qui contient tous les **langages finis** sur A et qui est stable par les **opérations rationnelles**:

- (i) somme (union); $L_1, L_2 \in \operatorname{Rat} A^* \Rightarrow L_1 + L_2 \in \operatorname{Rat} A^*$
- (ii) produit (concaténation); $L_1, L_2 \in \operatorname{Rat} A^* \Rightarrow L_1 \cdot L_2 \in \operatorname{Rat} A^*$
- (iii) itération (étoile de Kleene). $L \in \operatorname{Rat} A^* \Rightarrow L^* \in \operatorname{Rat} A^*$

Langages rationnels

Définition La famille Rat A^* des **langages rationnels** sur l'alphabet A est la plus petite famille de langages qui contient tous les **langages finis** sur A et qui est stable par les **opérations rationnelles** :

- (i) somme (union); $L_1, L_2 \in \operatorname{Rat} A^* \Rightarrow L_1 + L_2 \in \operatorname{Rat} A^*$
- (ii) produit (concaténation); $L_1, L_2 \in \operatorname{Rat} A^* \Rightarrow L_1 \cdot L_2 \in \operatorname{Rat} A^*$
- (iii) itération (étoile de Kleene). $L \in \operatorname{Rat} A^* \Rightarrow L^* \in \operatorname{Rat} A^*$

On peut énoncer un théorème fondamental de la théorie des langages.

Langages rationnels

Définition La famille Rat A^* des **langages rationnels** sur l'alphabet A est la plus petite famille de langages qui contient tous les **langages finis** sur A et qui est stable par les **opérations rationnelles**:

- (i) somme (union); $L_1, L_2 \in \operatorname{Rat} A^* \Rightarrow L_1 + L_2 \in \operatorname{Rat} A^*$
- (ii) produit (concaténation); $L_1, L_2 \in \operatorname{Rat} A^* \Rightarrow L_1 \cdot L_2 \in \operatorname{Rat} A^*$
- (iii) itération (étoile de Kleene). $L \in \operatorname{Rat} A^* \Rightarrow L^* \in \operatorname{Rat} A^*$

On peut énoncer un théorème fondamental de la théorie des langages.

Théorème de Kleene Un langage est rationnel si et seulement si il est reconnaissable.