

Théorie des langages

Julien BERNARD

Université de Franche-Comté – UFR Sciences et Technique
Licence Informatique – 3^è année

2016 – 2017

Première partie

Introduction – Alphabets, mots, langages

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- Langage
- Méthodologie

Plan

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- Langage
- Méthodologie

Votre enseignant

Qui suis-je ?

Qui suis-je ?

Julien BERNARD, Maître de Conférence (enseignant-chercheur)
`julien.bernard@univ-fcomte.fr`, Bureau 426C

Enseignement

- Responsable du semestre 1 (Starter) de la licence Informatique
- Cours : Bases de la programmation (L1), Publication web et scientifique (L1), Algorithmique (L2), Sécurité (L3), Théorie des Langages (L3)

Recherche

Optimisation dans les réseaux de capteurs

Plan

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- Langage
- Méthodologie

UE Théorie des Langages

Organisation

Équipe pédagogique

- Julien Bernard : CM, TD (julien.bernard@univ-fcomte.fr)
- Hana M Hemdi : TP (hana.m_hemdi@edu.univ-fcomte.fr)
- Guillaume Voiron : TP (guillaume.voiron@edu.univ-fcomte.fr)

Volume

- Cours : 12 x 1h30, jeudi 11h00
- TD : 12 x 1h30, vendredi 13h30 (Gr. 1) et 15h00 (Gr. 2)
- TP : 6 x 3h00

Évaluation

- 2 devoirs surveillés
- un projet en TP

UE Théorie des Langues

Comment ça marche ?

Mode d'emploi

- 1 Prenez des notes ! Posez des questions !
- 2 Comprendre plutôt qu'apprendre
- 3 Le but de cette UE n'est pas d'avoir une note !

Niveau d'importance des transparents

	trivial	pour votre culture
★	intéressant	pour votre compréhension
★★	important	pour votre savoir
★★★	vital	pour votre survie

Note : les contrôles portent sur *tous* les transparents !

UE Théorie des Langages

Contenu pédagogique

Objectif

Comprendre la théorie et les outils de la théorie des langages

- Alphabets, mots, langages
- Grammaires
- Langages réguliers
- Automates d'états finis
- Expressions régulières
- Langages algébriques
- Automates à piles
- Machines de Turing

UE Théorie des Langages

Bibliographie



P. Wolper

Introduction à la calculabilité.

2006, Dunod



P. Séébold.

Théorie des automates.

2009, Vuibert



J.M. Autebert

Théorie des langages et des automates.

1994, Masson



J. Hopcroft, J. Ullman

Introduction to Automata Theory, Languages and Compilation

1979, Addison-Wesley

Plan

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- Langage
- Méthodologie

Bref historique

Historique

- Notion de langage formel, Noam Chomsky, début des années 1950
 - Origine dans la linguistique
 - Étude des langues naturelles
 - Traitement automatique (exemple : traduction)
- Hiérarchie de Chomsky, 1956
 - Classification des langages selon leur pouvoir d'expression
- Outil important en informatique !

Étude des langages formels

Niveaux d'études

Niveaux d'études des langages

Deux points de vue :

- Du **locuteur**. Le problème est de savoir engendrer les phrases (mots) du langage → Notion de **grammaire**
- De l'**auditeur**. Le problème est de savoir reconnaître les phrases (mots) du langage → Notion de **reconnaisseur** (automate)

Étude des langages formels

Buts

Buts de l'étude des langages

- Évaluer et classer les langages
 - Caractériser les langages
 - Trouver des grammaires
 - Trouver des reconnaisseurs
- Développer des algorithmes
 - Compilation des langages informatiques
 - Reconnaissance de la parole

Plan

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- Langage
- Méthodologie

Alphabet



Définition (Alphabet)

Un **alphabet** est un ensemble fini de symboles appelés **lettres**.

Remarque

Un alphabet est aussi appelé **vocabulaire**.

Exemples (Alphabet)

- $A = \{0, 1\}$
- $\Sigma = \{a, b, c\}$
- $\Theta = \{if, then, else, a, b\}$
- $F = \{\rightarrow, \leftarrow, \uparrow, \downarrow\}$

Mot



Définition (Mot)

Un **mot** sur l'alphabet A est une suite *finie* et *ordonnée*, éventuellement vide, de lettres de A . Le **mot vide** est toujours noté ε .

Exemples (Mot)

aba et $abbaccb$ sont deux mots sur l'alphabet $A = \{a, b, c\}$.

Notation

Soit w un mot constitué de k lettres sur l'alphabet A , on notera :

$$w = w_1 \cdots w_k$$

Longueur d'un mot



Définition (Longueur d'un mot)

La **longueur d'un mot** w est le nombre de lettres constituant le mot w . Elle est notée $|w|$. Le mot vide a une longueur de 0.

Exemples (Longueur d'un mot)

$|aba| = 3$, $|abbacbb| = 7$, $|\varepsilon| = 0$

Remarque

De nombreuses propriétés sur les mots se montreront par récurrence sur la longueur des mots.

Ensemble des mots



Définition (Ensemble des mots)

L'**ensemble des mots non-vide** sur un alphabet A est noté A^+ .

$$A^+ = \{w = w_1 \cdots w_n, n > 0\}$$

L'**ensemble des mots** sur un alphabet A est noté A^* .

$$A^* = \{\varepsilon\} \cup A^+ = \{w = w_1 \cdots w_n, n \geq 0\}$$

Produit de mots



Définition (Produit de mots)

Soient A un alphabet et $x, y \in A^*$ deux mots sur l'alphabet A de longueur respective n et m . On définit le **produit** w de x et y noté $x \cdot y$ par :

$$w = x \cdot y = x_1 \cdots x_n \cdot y_1 \cdots y_m = x_1 \cdots x_n y_1 \cdots y_m$$

Remarque

Le produit est aussi appelé **concaténation**.

Exemple (Produit de mots)

$$aba \cdot ab = abaab$$

Monoïde $(A^*, \cdot, \varepsilon)$



Proposition (Monoïde $(A^*, \cdot, \varepsilon)$)

A^ munie de l'opération produit d'élément neutre ε est un monoïde.*

Démonstration.

- Le produit est une loi interne : $x \cdot y \in A^*$
- Le produit est associatif : $x \cdot (y \cdot z) = (x \cdot y) \cdot z$
- ε est l'élément neutre du produit : $x \cdot \varepsilon = \varepsilon \cdot x = x$



Remarque

Le produit n'est pas commutatif : $x \cdot y \neq y \cdot x$

Puissance d'un mot



Définition (Puissance d'un mot)

Soient A un alphabet, et $w \in A^*$. La **puissance** d'un mot, noté w^n est définie par :

$$w^n = \begin{cases} \varepsilon & \text{si } n = 0 \\ w \cdot w^{n-1} & \text{si } n > 0 \end{cases}$$

Exemple (Puissance d'un mot)

Soit $A = \{a, b\}$ et $w = abb$, alors :

- $w^0 = \varepsilon$
- $w^1 = w = abb$
- $w^2 = w \cdot w = abbabb$
- $w^3 = w \cdot w^2 = abbabbabb$

Égalité de mots



Définition (Égalité de mots)

Deux mots sont **égaux** si et seulement s'ils sont de même longueur et s'ils ont des lettres identiques à des positionnements identiques.

Soit A un alphabet et $x = x_1 \cdots x_n, y = y_1 \cdots y_m \in A^*$, alors :

$$x = y \iff n = m \text{ et } \forall i \in [1, n], x_i = y_i$$

Plan

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- **Langage**
- Méthodologie

Langage



Définition (Langage)

Un **langage** L sur un alphabet A est un sous-ensemble de A^* . C'est un ensemble de mots sur l'alphabet A .

Exemples (Langage)

Soit $A = \{a, b\}$ un alphabet :

- $L = \emptyset$ est un langage appelé **langage vide**
- $L = \{\varepsilon\}$ est un langage appelé **langage unité**
- $L = \{a, ab, abb\}$ est un langage
- $L = \{a^n, n \geq 0\}$ est un langage

Égalité de langage



Définition (Égalité de langage)

Deux langages L_1 et L_2 sont **égaux**, noté $L_1 = L_2$ si et seulement si $L_1 \subseteq L_2$ et $L_2 \subseteq L_1$.

Remarque

Cette définition donne une manière de prouver l'égalité de deux langages.

Complémentaire d'un langage



Définition (Complémentaire d'un langage)

Soit L un langage sur l'alphabet A , le **complémentaire** de L , noté \bar{L} est le langage défini par :

$$\bar{L} = \{w \in A^*, w \notin L\}$$

Exemple (Complémentaire d'un langage)

Soit $L = \{w \in A^*, |w| \equiv 0 \pmod{2}\}$, alors, $\bar{L} = \{w \in A^*, |w| \equiv 1 \pmod{2}\}$

Union de langages



Définition (Union de deux langages)

Soient L_1 et L_2 deux langages sur l'alphabet A , l'**union** de L_1 et L_2 , notée $L_1 \cup L_2$ est définie par :

$$L_1 \cup L_2 = \{w \in A^*, w \in L_1 \text{ ou } w \in L_2\}$$

Propriétés

L'union est :

- Associative : $L_1 \cup (L_2 \cup L_3) = (L_1 \cup L_2) \cup L_3$
- Commutative : $L_1 \cup L_2 = L_2 \cup L_1$
- Idempotente : $L \cup L = L$
- Élément neutre \emptyset : $L \cup \emptyset = \emptyset \cup L = L$

Intersection de langages



Définition (Intersection de deux langages)

Soient L_1 et L_2 deux langages sur l'alphabet A , l'**intersection** de L_1 et L_2 , notée $L_1 \cap L_2$ est définie par :

$$L_1 \cap L_2 = \{w \in A^*, w \in L_1 \text{ et } w \in L_2\}$$

Propriétés

L'intersection est :

- Associative : $L_1 \cap (L_2 \cap L_3) = (L_1 \cap L_2) \cap L_3$
- Commutative : $L_1 \cap L_2 = L_2 \cap L_1$
- Idempotente : $L \cap L = L$
- Élément neutre A^* : $L \cap A^* = A^* \cap L = L$

Différence de langages



Définition (Différences de deux langages)

Soient L_1 et L_2 deux langages sur l'alphabet A , la **différence** de L_1 et L_2 , notée $L_1 \setminus L_2$ est définie par :

$$L_1 \setminus L_2 = \{w \in A^*, w \in L_1 \text{ et } w \notin L_2\}$$

Opérations ensemblistes sur les langages

Exemples (Opérations ensemblistes sur les langages)

Soit $A = \{a, b\}$ un alphabet.

Soit $L_1 = \{a, ab\}$ et $L_2 = \{ab, ba\}$ deux langages sur A .

- $L_1 \cup L_2 = L_2 \cup L_1 = \{a, ab, ba\}$
- $L_1 \cap L_2 = L_2 \cap L_1 = \{ab\}$
- $L_1 \setminus L_2 = \{a\}$
- $L_2 \setminus L_1 = \{ba\}$

Produit de langages



Définition (Produit de deux langages)

Soient A un alphabet et $L_1, L_2 \subseteq A^*$ deux langages sur l'alphabet A . On définit le **produit** L de L_1 et L_2 noté $L_1.L_2$ par :

$$L = L_1.L_2 = \{u_1 \cdot u_2, u_1 \in L_1, u_2 \in L_2\}$$

Remarque

- Le produit est aussi appelé **concaténation**.
- Attention à ne pas confondre avec le produit cartésien (noté \times).

Produit de langage

Exemple (Produit de deux langages)

Soit $A = \{a, b\}$ un alphabet.

Soit $L_1 = \{\varepsilon, a, ab\}$ et $L_2 = \{b, ba\}$ deux langages sur A .

- $L_1.L_2 = \{b, ba, ab, aba, abb, abba\}$
- $L_2.L_1 = \{b, ba, bab, ba, baa, baab\} = \{b, ba, bab, baa, baab\}$

→ $L_1.L_2 \neq L_2.L_1$

Propriétés du produit de langage



Proposition (Distributivité du produit par rapport à l'union)

Le produit de langage est distributif par rapport à l'union. Soient A un alphabet et $L_1, L_2, L_3 \subseteq A^$, alors :*

$$L_1.(L_2 \cup L_3) = (L_1.L_2) \cup (L_1.L_3) \text{ et } (L_1 \cup L_2).L_3 = (L_1.L_3) \cup (L_2.L_3)$$

Remarque importante

Le produit de langage n'est pas distributif par rapport à l'intersection. Plus précisément, on a :

$$L_1.(L_2 \cap L_3) \subseteq (L_1.L_2) \cap (L_1.L_3) \text{ et } (L_1 \cap L_2).L_3 \subseteq L_1.L_3 \cap L_2.L_3$$

Propriétés du produit de langage



Distributivité du produit par rapport à l'union.

- ① Soit $w \in L_1.(L_2 \cup L_3)$, montrons que $w \in (L_1.L_2) \cup (L_1.L_3)$.
 $\exists w_1 \in L_1, w' \in L_2 \cup L_3, w = w_1 \cdot w'$. Donc $w' \in L_2$ ou $w' \in L_3$.
Si $w' \in L_2$, alors $w = w_1 \cdot w' \in L_1.L_2$.
Si $w' \in L_3$, alors $w = w_1 \cdot w' \in L_1.L_3$.
Donc, $w \in (L_1.L_2) \cup (L_1.L_3)$.
Donc $L_1.(L_2 \cup L_3) \subseteq (L_1.L_2) \cup (L_1.L_3)$.
- ② Soit $w \in (L_1.L_2) \cup (L_1.L_3)$, montrons que $w \in L_1.(L_2 \cup L_3)$.
 $w \in L_1.L_2$ ou $w \in L_1.L_3$.
Si $w \in L_1.L_i, i \in \{2, 3\}$ alors $\exists w_1 \in L_1, w_i \in L_i, w = w_1 \cdot w_i$.
Donc $w \in L_1.(L_2 \cup L_3)$.
Donc, $(L_1.L_2) \cup (L_1.L_3) \subseteq L_1.(L_2 \cup L_3)$

Donc $(L_1.L_2) \cup (L_1.L_3) = L_1.(L_2 \cup L_3)$



Puissance d'un langage



Définition (Puissance d'un langage)

Soient A un alphabet, et $L \subseteq A^*$. La **puissance** d'un langage, noté L^n est définie par :

$$L^n = \begin{cases} \{\varepsilon\} & \text{si } n = 0 \\ L.L^{n-1} & \text{si } n > 0 \end{cases}$$

Itéré d'un langage



Définition (Itéré d'un langage)

L'**itéré strict** d'un langage L , noté L^+ , est défini par :

$$L^+ = \bigcup_{i>0} L^i$$

L'**itéré** d'un langage L , appelé aussi l'**étoile de Kleene**, noté L^* , est défini par :

$$L^* = \bigcup_{i \geq 0} L^i = \{\varepsilon\} \cup L^+$$

Proposition

Soit L un langage, alors on a :

$$L^+ = L.L^* = L^*.L$$

Plan

1 Introduction

- À propos de votre enseignant
- À propos du cours Théorie des Langages

2 Alphabets, mots, langages

- Contexte
- Alphabets et mots
- Langage
- Méthodologie

Appartenance d'un élément à un ensemble



Appartenance d'un élément à un ensemble

Soit X un ensemble défini par une propriété P_X :

$$X = \{x, P_X(x)\}$$

.

Pour prouver qu'un élément y appartient à l'ensemble X , il suffit de montrer qu'il satisfait la propriété P_X .

Inclusion d'un ensemble dans un autre



Inclusion d'un ensemble dans un autre

Soit X et Y deux ensembles. Pour prouver que l'ensemble X est inclus dans l'ensemble Y , on cherche à montrer :

$$\forall x \in X, x \in Y$$

- On considère un élément $x \in X$, c'est-à-dire qu'il satisfait P_X .
- On montre que l'élément x est dans Y , c'est-à-dire qu'il satisfait P_Y .
- On conclue.

Égalité de deux ensembles



Égalité de deux ensembles

Soit X et Y deux ensembles. Pour prouver que les ensembles X et Y sont égaux, on cherche à montrer :

$$X \subseteq Y \text{ et } Y \subseteq X$$

Deuxième partie

Grammaires

- 3 Grammaires
 - Définitions
 - Réécriture, dérivation et langage engendré
 - Arbre de dérivation
 - Hiérarchie de Chomsky

Plan

3 Grammaires

- Définitions

- Réécriture, dérivation et langage engendré
- Arbre de dérivation
- Hiérarchie de Chomsky

Grammaire

Principe



Principe d'une grammaire

Ensemble de règles pour générer les mots du langage

- On part d'un symbole spécial appelé l'**axiome** ou la **source**
- On applique des **règles de réécriture**
 - Remplacement d'une séquence de symboles par une autre séquence
- On génère des mots

Grammaire

Exemple introductif



Exemple introductif

- On considère la phrase suivante :

la vieille dame regarde la petite fille

→ Peut-on construire une grammaire qui génère cette phrase ?

- Alphabet : { la, vieille, petite, dame, fille, regarde }
- Structure de la phrase :
 - Un groupe sujet (article, adjectif, nom)
 - Un verbe
 - Un groupe complément d'objet (article, adjectif, nom)

Grammaire

Exemple introductif



Règles de production

- 1 $\langle \text{Phrase} \rangle \rightarrow \langle \text{Sujet} \rangle \langle \text{Verbe} \rangle \langle \text{Complément} \rangle$
- 2 $\langle \text{Sujet} \rangle \rightarrow \langle \text{Groupe Nominal} \rangle$
- 3 $\langle \text{Complément} \rangle \rightarrow \langle \text{Groupe Nominal} \rangle$
- 4 $\langle \text{Groupe Nominal} \rangle \rightarrow \langle \text{Article} \rangle \langle \text{Nom} \rangle$
- 5 $\langle \text{Groupe Nominal} \rangle \rightarrow \langle \text{Article} \rangle \langle \text{Adjectif} \rangle \langle \text{Nom} \rangle$
- 6 $\langle \text{Article} \rangle \rightarrow \text{la}$
- 7 $\langle \text{Nom} \rangle \rightarrow \text{dame} \mid \text{fille}$
- 8 $\langle \text{Adjectif} \rangle \rightarrow \text{vieille} \mid \text{petite}$
- 9 $\langle \text{Verbe} \rangle \rightarrow \text{regarde}$

Définition (Grammaire)

Une **grammaire** G est un quadruplet (N, T, X, R) où :

- N est l'ensemble des **symboles non-terminaux**
- T est l'ensemble des **symboles terminaux**
- $V = N \cup T$ est le **vocabulaire** de la grammaire ($N \cap T = \emptyset$)
- $X \in N$ est l'**axiome**
- R est un ensemble de **règles de production** de la forme :

$$\alpha \rightarrow \beta, \alpha \in V^+, \beta \in V^*$$

Remarques

- Généralement, les symboles non-terminaux sont écrits en majuscules
- Généralement, les symboles terminaux sont écrits en minuscules
- $\alpha \rightarrow \beta$ signifie que α peut être remplacé par β

Notation

Plusieurs règles qui ont le même membre gauche peuvent être factorisées.

- $\alpha \rightarrow \beta$
- $\alpha \rightarrow \gamma$

est équivalent à :

- $\alpha \rightarrow \beta \mid \gamma$

Exemples de grammaire

Exemple (Grammaire)

$G = (\{E, T, F\}, \{+, \times, (,), a, b\}, E, R)$ avec R :

- $E \rightarrow E + T \mid T$
- $T \rightarrow T \times F \mid F$
- $F \rightarrow (E) \mid a \mid b$

Exemples de grammaire

Exemple (Grammaire)

$G = (\{S\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow aSa \mid SbS \mid \varepsilon$

Exemples de grammaire

Exemple (Grammaire)

$G = (\{S, B, C\}, \{a, b, c\}, S, R)$ avec R :

- $S \rightarrow aSBC \mid \varepsilon$
- $CB \rightarrow BC$
- $aB \rightarrow ab$
- $bB \rightarrow bb$
- $bC \rightarrow bc$
- $cC \rightarrow cc$

Plan

3 Grammaires

- Définitions
- Réécriture, dérivation et langage engendré
- Arbre de dérivation
- Hiérarchie de Chomsky

Réécriture



Définition (Réécriture)

Soit $G = (N, T, X, R)$ une grammaire, $u \in V^+$, $v \in V^*$.

La **réécriture** de u en v par G , notée $u \rightarrow v$, est définie par :

- $u = \alpha \cdot u' \cdot \beta, v = \alpha \cdot v' \cdot \beta$ avec $\alpha, \beta \in V^*$
- $(u' \rightarrow v') \in R$

Réécriture

Exemple (Réécriture)

Soit $G = (\{E, T, F\}, \{+, \times, (,), a, b\}, E, R)$ avec R :

- $E \rightarrow E + T \mid T$
- $T \rightarrow T \times F \mid F$
- $F \rightarrow (E) \mid a \mid b$

On peut effectuer des réécritures jusqu'au mot $a + b \times a$:

$$\begin{aligned} E &\rightarrow E + T \rightarrow T + T \rightarrow F + T \rightarrow F + T \times F \\ &\rightarrow a + T \times F \rightarrow a + F \times F \rightarrow a + F \times a \rightarrow a + b \times a \end{aligned}$$

Dérivation



Définition (Dérivation)

Soit $G = (N, T, X, R)$ une grammaire, $u \in V^+$, $v \in V^*$.

La **dérivation** de u en v par G , notée $u \rightarrow^* v$, est définie par :

- $\exists k > 0$ et $\exists w_0, \dots, w_k \in V^*$, avec $w_0 = u$ et $w_k = v$
- $w_i \rightarrow w_{i+1}$ pour tout $0 \leq i < k$

Exemple (Dérivation)

Avec la grammaire précédente, on a :

$$E \rightarrow^* F + T \times F \rightarrow^* a + b \times a$$

Mot engendré par une grammaire



Définition (Mot engendré par une grammaire)

Soit $G = (N, T, X, R)$ une grammaire,

$u \in T^*$ (symboles terminaux) est un **mot engendré par la grammaire G** s'il peut être dérivé depuis l'axiome X , c'est-à-dire $X \rightarrow^* u$.

Exemple (Mot engendré par une grammaire)

$a + b \times a$ est un mot engendré par la grammaire G précédente.

Langage engendré par une grammaire



Définition (Langage engendré par une grammaire)

Soit $G = (N, T, X, R)$ une grammaire,

Le **langage engendré par la grammaire** G , noté $\mathcal{L}(G)$, est l'ensemble des mots engendrés par G .

$$\mathcal{L}(G) = \{u \in T^*, X \rightarrow^* u\}$$

Exemple (Langage engendré par une grammaire)

Soit $G = (\{X\}, \{a, b\}, X, R)$ avec R :

- $X \rightarrow aXbX \mid bXaX \mid \varepsilon$

Le langage $\mathcal{L}(G)$ engendré par G est le langage L des mots qui contiennent autant de a que de b .

$$\mathcal{L}(G) = \{u \in T^*, |u|_a = |u|_b\}$$

Dérivation la plus à gauche



Définition (Dérivation la plus à gauche)

Soit $G = (N, T, X, R)$ une grammaire, et $w \in \mathcal{L}(G)$, la dérivation $S \rightarrow^* w$ est la **dérivation la plus à gauche** si, à chaque étape de la dérivation, c'est le symbole non-terminal le plus à gauche qui est dérivé.

Plan

3 Grammaires

- Définitions
- Réécriture, dérivation et langage engendré
- **Arbre de dérivation**
- Hiérarchie de Chomsky

Arbre de dérivation



Définition (Arbre de dérivation)

Soit $G = (N, T, X, R)$ une grammaire et $w \in \mathcal{L}(G)$.

L'**arbre de dérivation** du mot w est un arbre où :

- la racine est X
- les feuilles sont étiquetées par des éléments terminaux de T
- les nœuds sont étiquetés par des éléments non-terminaux de N
- si un nœud est étiqueté Y et ses fils sont étiquetés Z_1, \dots, Z_k dans cet ordre, alors il existe une règle $Y \rightarrow Z_1 \dots Z_k$ dans R
- la lecture des feuilles de gauche à droite donne le mot w

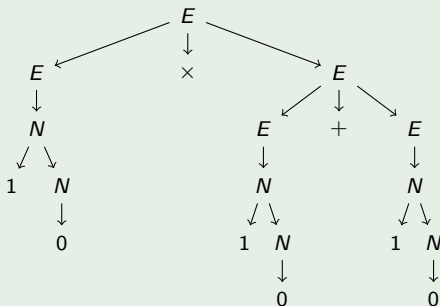
Arbre de dérivation

Exemple (Arbre de dérivation)

Soit $G = (\{E, N\}, \{+, \times, 0, 1\}, E, R)$ avec R :

- $E \rightarrow E + E \mid E \times E \mid N$
- $N \rightarrow 0 \mid 1 \mid 0N \mid 1N$

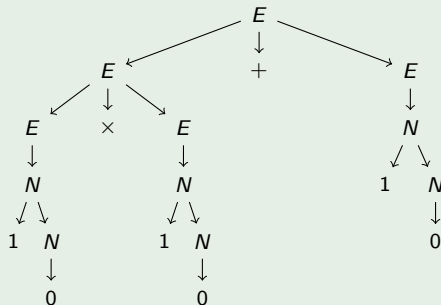
Le mot $10 \times 10 + 10$ a pour arbre de dérivation :



Arbre de dérivation

Exemple (Arbre de dérivation)

Le mot $10 \times 10 + 10$ a également pour arbre de dérivation :



Grammaire ambiguë



Définition (Grammaire ambiguë)

Une grammaire G est **ambiguë** s'il existe un mot de $\mathcal{L}(G)$ qui a au moins deux arbres de dérivation, c'est-à-dire deux dérivations la plus à gauche.

Exemple (Grammaire ambiguë)

La grammaire $G = (\{E, N\}, \{+, \times, 0, 1\}, E, R)$ avec R :

- $E \rightarrow E + E \mid E \times E \mid N$
- $N \rightarrow 0 \mid 1 \mid 0N \mid 1N$

est ambiguë.

Plan

3 Grammaires

- Définitions
- Réécriture, dérivation et langage engendré
- Arbre de dérivation
- Hiérarchie de Chomsky

Grammaire générale (type 0)



Définition (Grammaire générale (type 0))

Une **grammaire de type 0** ou **grammaire générale** est une grammaire sans restriction sur la forme des règles.

Définition (Langage général)

Un **langage général** est un langage engendré par une grammaire générale.

Grammaire contextuelle (type 1)



Définition (Grammaire contextuelle (type 1))

Une **grammaire de type 1** ou **grammaire contextuelle** est une grammaire où les règles sont de la forme :

$$\alpha A \beta \rightarrow \alpha w \beta$$

avec $\alpha, \beta \in V^*$, $w \in V^+$ et $A \in N$. Le symbole A est remplacé par w si on a le contexte α à gauche et β à droite.

Définition (Langage contextuel (type 1))

Un **langage contextuel** est un langage engendré par une grammaire contextuelle.

Grammaire contextuelle (type 1)

Exemple (Grammaire contextuelle (type 1))

La grammaire $G = (\{S, B, C, H\}, \{a, b, c\}, S, R)$ avec R :

- $S \rightarrow aSBC$
- $S \rightarrow aBC$
- $CB \rightarrow HB$
- $HB \rightarrow HC$
- $HC \rightarrow BC$
- $aB \rightarrow ab$
- $bB \rightarrow bb$
- $bC \rightarrow bc$
- $cC \rightarrow cc$

engendre le langage $L = \{a^n b^n c^n, n \geq 1\}$

Grammaire contextuelle (type 1)



Définition (Grammaire croissante)

Une **grammaire croissante** est une grammaire où les règles sont de la forme :

$$\alpha \rightarrow \beta$$

avec $\alpha, \beta \in V^*$ et $|\alpha| \leq |\beta|$

Proposition

Les langages engendrés par les grammaires croissantes sont les langages contextuels.

Grammaire contextuelle (type 1)

Exemple (Grammaire croissante)

La grammaire $G = (\{S, B\}, \{a, b, c\}, S, R)$ avec R :

- $S \rightarrow abc \mid aSBc$
- $cB \rightarrow Bc$
- $bB \rightarrow bb$

engendre le langage $L = \{a^n b^n c^n, n \geq 1\}$

Grammaire algébrique (type 2)



Définition (Grammaire algébrique (type 2))

Une **grammaire de type 2** ou **grammaire algébrique** ou **grammaire hors contexte** est une grammaire où les règles sont de la forme :

$$A \rightarrow \alpha$$

avec $A \in N$ et $\alpha \in V^*$, la partie gauche est réduite à un non-terminal.

Définition (Langage algébrique (type 2))

Un **langage algébrique** est un langage engendré par une grammaire algébrique.

Grammaire algébrique (type 2)

Exemple (Grammaire algébrique (type 2))

La grammaire $G = (\{S\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow \varepsilon \mid aSb$

engendre le langage $L = \{a^n b^n, n \geq 0\}$

Grammaire régulière (type 3)



Définition (Grammaire régulière (type 3))

Une **grammaire de type 3** ou **grammaire régulière** est une grammaire où les règles sont de la forme :

$$A \rightarrow a \text{ ou } A \rightarrow aB$$

avec $A, B \in N$ et $a \in T$.

Définition (Langage régulier (type 3))

Un **langage régulier** est un langage engendré par une grammaire régulière.

Grammaire régulière (type 3)

Exemple (Grammaire régulière (type 3))

La grammaire $G = (\{S, A, B\}, \{a, b\}, S, R)$ avec R :

- $S \rightarrow aA$
- $A \rightarrow bA \mid bB$
- $B \rightarrow a$

engendre le langage $L = \{ab^n a, n \geq 1\}$

Hiérarchie de Chomsky



Hiérarchie de Chomsky

Type	Grammaire	Reconnaisseur
0	Générale	Machine de Turing
1	Contextuelle	
2	Algébrique	Automate à pile
3	Régulière	Automate

Troisième partie

Automates d'états finis

- 4 Automates d'états finis
 - Automates finis déterministes
 - Représentations d'un automate
 - Automates équivalents et complets
 - Automates finis non-déterministes

Plan

- 4 Automates d'états finis
 - Automates finis déterministes
 - Représentations d'un automate
 - Automates équivalents et complets
 - Automates finis non-déterministes

Automate fini déterministe



Définition (Automate fini déterministe)

Un **automate fini déterministe** \mathcal{A} sur un alphabet A est un quadruplet (Q, q_0, δ, Q_F) où :

- Q est un ensemble fini d'états
- $q_0 \in Q$ est un état initial
- $\delta : Q \times A \rightarrow Q$ est la fonction de transition
- $Q_F \subseteq Q$ est un ensemble d'états finaux

Remarque

La fonction de transition δ peut être partielle, c'est-à-dire non-définie sur tout l'ensemble $Q \times A$.

Automate fini déterministe

Exemple (Automate fini déterministe)

$\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Fonction de transition



Fonction de transition étendue δ

On étend la fonction $\delta : Q \times A^* \rightarrow Q$ aux mots sur l'alphabet A de la manière suivante :

- $\delta(q, \varepsilon) = q$
- $\delta(q, a \cdot \alpha) = \delta(\delta(q, a), \alpha), a \in A, \alpha \in A^*$

Notation alternative

La fonction $\delta(q, a)$ est parfois notée $q \cdot a$. Les deux propriétés précédentes s'écrivent alors :

- $q \cdot \varepsilon = q$
- $q \cdot (a \cdot \alpha) = (q \cdot a) \cdot \alpha$

Fonction de transition

Exemple (Fonction de transition)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors, par exemple, on a :

- $\delta(q_0, ba) = q_2$
- $\delta(q_0, baaaaaaaa) = q_2$
- $\delta(q_0, baaaaaab) = q_3$
- $\delta(q_2, aab) = q_3$
- $\delta(q_0, bb)$ n'est pas défini parce que $\delta(q_1, b)$ n'est pas défini

Dérivation en une étape



Définition (Configuration)

Une **configuration** est une paire (q, w) , où $q \in Q, w \in A^*$

Définition (Dérivation en une étape)

Soit \mathcal{A} un automate et (q, w) et (q', w') deux configurations. La configuration (q', w') est **dérivable en une étape** de la configuration (q, w) par \mathcal{A} , noté $(q, w) \mapsto (q', w')$ si :

- $w = x \cdot w'$ avec $x \in A$
- \mathcal{A} est dans l'état q
- $q' = \delta(q, a)$

Remarque

On dit qu'on «lit» la lettre a .

Dérivation en une étape

Exemple (Dérivation en une étape)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors, on a les dérivations suivantes :

$$(q_0, baaab) \mapsto (q_1, aaab) \mapsto (q_2, aab) \mapsto (q_2, ab) \mapsto (q_2, b) \mapsto (q_3, \varepsilon)$$

Dérivation



Définition (Dérivation)

Soit \mathcal{A} un automate et (q, w) et (q', w') deux configurations. La configuration (q', w') est **dérivable** de la configuration (q, w) par \mathcal{A} , noté $(q, w) \mapsto^* (q', w')$ si :

- $\exists k \geq 0$ et k configurations $(q_i, w_i), 1 \leq i \leq k$
avec $(q_0, w_0) = (q, w)$ et $(q_k, w_k) = (q', w')$
- $(q_i, w_i) \mapsto (q_{i+1}, w_{i+1}), 1 \leq i < k$

Dérivation

Exemple (Dérivation)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors, on a les dérivations suivantes :

$$(q_0, baaab) \mapsto^* (q_2, aab) \mapsto^* (q_3, \varepsilon)$$

Mot accepté par un automate



Définition (Mot accepté)

Un mot w est **accepté** par un automate si :

$$(q_0, w) \mapsto^* (q, \varepsilon), q \in Q_F$$

Définition (Mot accepté)

Un mot w est **accepté** par un automate si :

$$\delta(q_0, w) = q, q \in Q_F$$

Remarque

On dit aussi que w est **reconnu** par un automate.

Mot accepté par un automate

Exemple (Mot accepté par un automate)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors, les mots suivants sont acceptés par l'automate :

- *bab*
- *baaaaab*

Les mots suivants ne sont pas acceptés par l'automate :

- *baaa*
- *bb*

Langage accepté par un automate



Définition (Langage accepté)

Le **langage accepté** par l'automate \mathcal{A} , noté $\mathcal{L}(\mathcal{A})$, est défini par :

$$\mathcal{L}(\mathcal{A}) = \{w \in A^*, (q_0, w) \mapsto^* (q, \varepsilon), q \in Q_F\}$$

Définition (Langage reconnaissable)

Un **langage reconnaissable** L est un langage tel qu'il existe un automate \mathcal{A} qui accepte le langage L :

$$L = \mathcal{L}(\mathcal{A})$$

Langage accepté par un automate

Exemple (Langage accepté par un automate)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors, le langage accepté par l'automate est :

$$\mathcal{L}(\mathcal{A}) = \{baa^n b, n \geq 0\} = \{ba^n b, n \geq 1\}$$

Plan

- 4 Automates d'états finis
 - Automates finis déterministes
 - Représentations d'un automate
 - Automates équivalents et complets
 - Automates finis non-déterministes

Représentations d'un automate



Représentations d'un automate

- Table de transition d'état
- Diagramme d'états-transitions

Représentation par une table



Table de transition

Une table de transition d'état est une table à deux dimensions avec :

- l'ensemble des états verticalement
- l'ensemble des lettres de l'alphabet horizontalement

Chaque case contient l'état suivant correspondant à l'état actuel et la lettre de l'alphabet.

Table de transition

Exemple (Table de transition)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors, la table de transition de l'automate est :

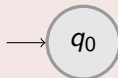
	a	b
q_0		q_1
q_1	q_2	
q_2	q_2	q_3
q_3		

Représentation par un diagramme

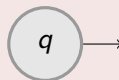


Notations

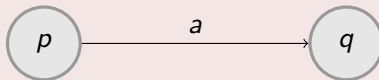
- État initial



- État final (deux notations)



- Transition entre l'état p et q : $\delta(p, a) = q$



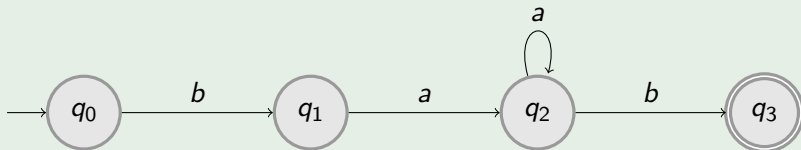
Représentation par un diagramme

Exemple (Diagramme)

Soit $\mathcal{A} = (\{q_0, q_1, q_2, q_3\}, q_0, \delta, \{q_3\})$ avec

$$\delta(q_0, b) = q_1, \delta(q_1, a) = q_2, \delta(q_2, a) = q_2, \delta(q_2, b) = q_3$$

Alors le diagramme d'états-transitions de l'automate est :



Plan

- 4 Automates d'états finis
 - Automates finis déterministes
 - Représentations d'un automate
 - Automates équivalents et complets
 - Automates finis non-déterministes

Automates équivalents



Définition (Automates équivalents)

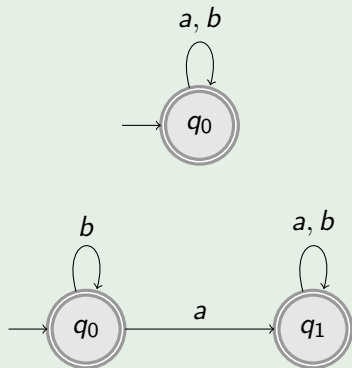
Deux automates \mathcal{A}_1 et \mathcal{A}_2 sont **équivalents** s'ils reconnaissent le même langage :

$$\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$$

Automates équivalents

Exemple (Automates équivalents)

Les deux automates suivants reconnaissent tous les deux le langage A^* , ils sont donc équivalents :



État accessible, co-accessible, utile



Définition (État accessible)

Un état q est **accessible** s'il existe un chemin entre q_0 et q .

Définition (État accessible)

Un état q est **co-accessible** s'il existe un chemin entre q et $q_f \in Q_F$.

Définition (État utile)

Un état est **utile** s'il est à la fois accessible et co-accessible.

Automate complet

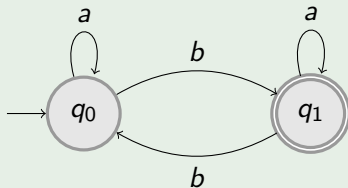


Définition (Automate complet)

Un automate est **complet** si pour tout état $q \in Q$, il existe une transition pour chaque lettre de l'alphabet A .

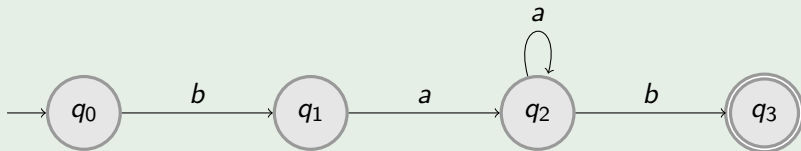
$$\forall q \in Q, \forall a \in A, \delta(q, a) \text{ est défini}$$

Exemple (Automate complet)



Automate complet

Exemple (Automate non-complet)



État puits et état poubelle



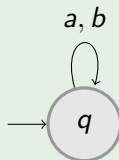
Définition (État puits)

Un **état puits** est un état $q \in Q$ pour lequel toutes les transitions sont de la forme $\delta(q, a) = q, a \in A$.

Définition (État poubelle)

Un **état poubelle** est un état puits non-final.

Exemple (État poubelle)



Automate complet



Proposition (Automate complet)

Pour tout automate fini, il existe un automate fini complet équivalent.

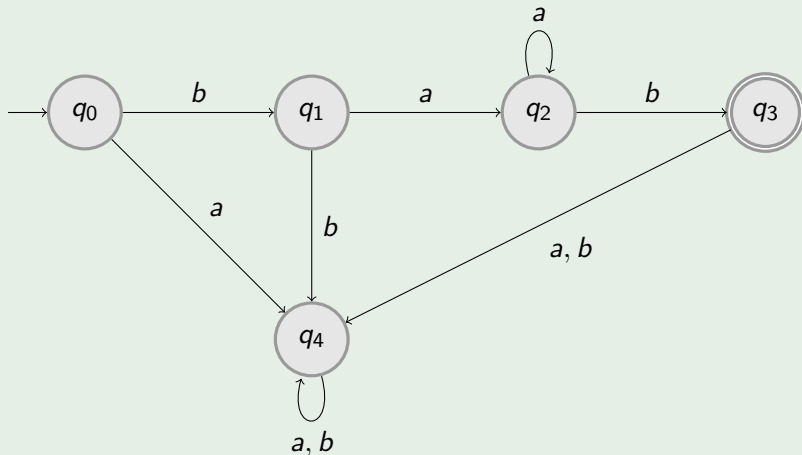
Démonstration.

Si l'automate n'est pas complet, on le complète en ajoutant un état poubelle.



Automate complet

Exemple (Automate complet)



Automate complet



Propriété

On peut toujours dériver un mot w sur un automate complet :

$$(q_0, w) \mapsto (q_1, w_1) \mapsto (q_2, w_2) \mapsto \dots \mapsto (q_n, \varepsilon)$$

On a deux possibilités :

- *Soit $q_n \in Q_F$, et le mot w est accepté par l'automate*
- *Soit $q_n \notin Q_F$, et le mot w n'est pas accepté par l'automate*

Plan

- 4 Automates d'états finis
 - Automates finis déterministes
 - Représentations d'un automate
 - Automates équivalents et complets
 - Automates finis non-déterministes

Automate fini non-déterministe



Définition (Automate fini non-déterministe)

Un **automate fini non-déterministe** \mathcal{A} sur un alphabet A est un quadruplet (Q, Q_I, Δ, Q_F) où :

- Q est un ensemble fini d'états
- $Q_I \subseteq Q$ est un ensemble d'état initiaux
- $\Delta \subseteq (Q \times A \cup \{\varepsilon\} \times Q)$ est une relation de transition
- $Q_F \subseteq Q$ est un ensemble d'états finaux

Automate fini non-déterministe

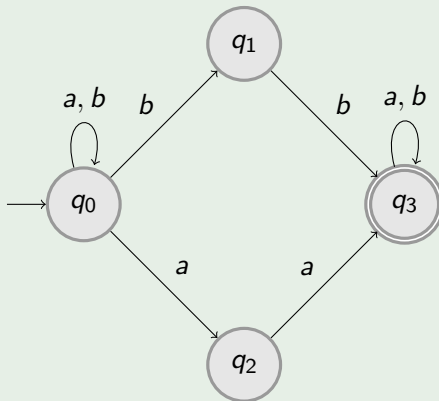


Différences entre automate fini déterministe et non-déterministe

- Il peut y avoir plusieurs états initiaux
- On n'a plus une fonction de transition mais une relation de transition
- Il peut y avoir des ε -transitions

Automate fini non-déterministe

Exemple (Automate fini non-déterministe)



Détermination



Théorème (Détermination)

Pour tout automate non-déterministe \mathcal{A}_N , il existe un automate fini déterministe \mathcal{A}_D équivalent.

$$\mathcal{L}(\mathcal{A}_N) = \mathcal{L}(\mathcal{A}_D)$$

Preuve constructive

Pour montrer ce théorème, on établit une preuve constructive, c'est-à-dire on donne un algorithme qui permet de produire un automate déterministe équivalent : l'algorithme de détermination d'un automate.

Détermination



Détermination

Soit $\mathcal{A}_N = (Q, Q_I, \Delta, Q_F)$, on définit l'automate $\mathcal{A}_D = \{R, r_0, \delta, R_F\}$ de la manière suivante :

- $R \subseteq 2^Q$,
 R est l'ensemble des sous-ensembles de Q
- $r_0 = \{q, q \in Q_I\}$,
 r_0 est l'ensemble des états initiaux de Q
- $\delta(r_1, a) = r_2 \iff r_2 = \{q_2 \in Q, \exists q_1 \in r_1, (q_1, a, q_2) \in \Delta\}$,
 r_2 est l'ensemble des états d'arrivée d'une transition par a depuis tous les états de r_1
- $R_F = \{r \in R, \exists q \in Q_F, q \in r\}$,
 R_F est l'ensemble des sous-ensembles de Q qui contiennent au moins un état final de Q_F

Déterminisation



Proposition

Soit \mathcal{A}_D l'automate obtenu après application de l'algorithme précédent sur l'automate \mathcal{A}_N , alors :

- \mathcal{A}_D est déterministe
- $\mathcal{L}(\mathcal{A}_D) = \mathcal{L}(\mathcal{A}_N)$

Remarque

Si l'automate \mathcal{A}_N contient n états, l'automate \mathcal{A}_D obtenu par l'algorithme de déterminisation peut contenir jusqu'à 2^n états.

Déterminisation en pratique



Déterminisation en pratique

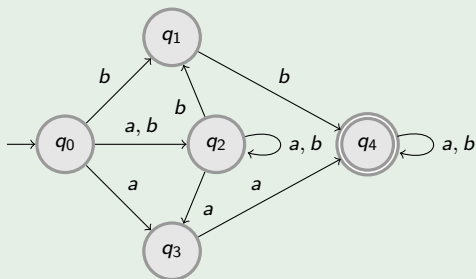
En pratique, on construit la table de transition de \mathcal{A}_D au fur et à mesure :

- ① On part de l'état initial de \mathcal{A}_D .
- ② Pour chaque nouvel ensemble d'états qui apparaît dans la table, on ajoute une ligne dans la table.
- ③ On recommence jusqu'à ce tous les ensembles aient été traités.

	a	b	\dots
$\{q_{i_1}, \dots, q_{i_k}\}$	$\{\dots\}$	$\{\dots\}$	\dots
\dots	\dots	\dots	\dots

Détermination

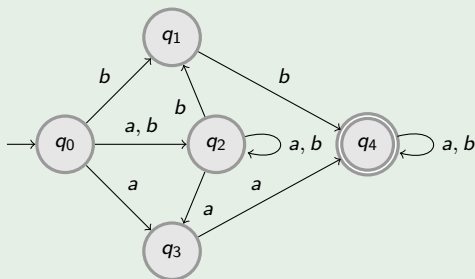
Exemple (Détermination)



	a	b
{q ₀ }		

Déterminisation

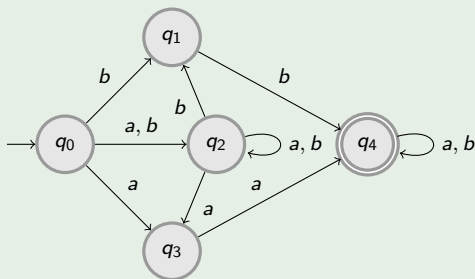
Exemple (Déterminisation)



	<i>a</i>	<i>b</i>
$\{q_0\}$	$\{q_2, q_3\}$	$\{q_1, q_2\}$
$\{q_2, q_3\}$		
$\{q_1, q_2\}$		

Déterminisation

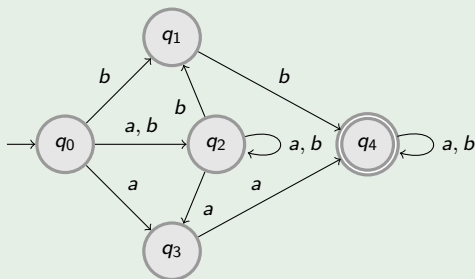
Exemple (Déterminisation)



	<i>a</i>	<i>b</i>
$\{q_0\}$	$\{q_2, q_3\}$	$\{q_1, q_2\}$
$\{q_2, q_3\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$		
$\{q_2, q_3, q_4\}$		

Déterminisation

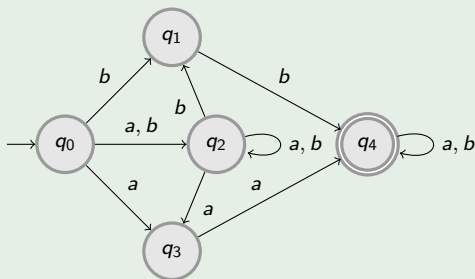
Exemple (Déterminisation)



	<i>a</i>	<i>b</i>
$\{q_0\}$	$\{q_2, q_3\}$	$\{q_1, q_2\}$
$\{q_2, q_3\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_2, q_3\}$	$\{q_1, q_2, q_4\}$
$\{q_2, q_3, q_4\}$		
$\{q_1, q_2, q_4\}$		

Déterminisation

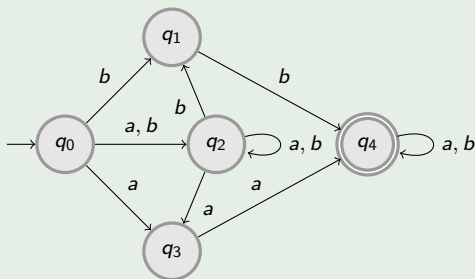
Exemple (Déterminisation)



	<i>a</i>	<i>b</i>
$\{q_0\}$	$\{q_2, q_3\}$	$\{q_1, q_2\}$
$\{q_2, q_3\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_2, q_3\}$	$\{q_1, q_2, q_4\}$
$\{q_2, q_3, q_4\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2, q_4\}$
$\{q_1, q_2, q_4\}$		

Déterminisation

Exemple (Déterminisation)

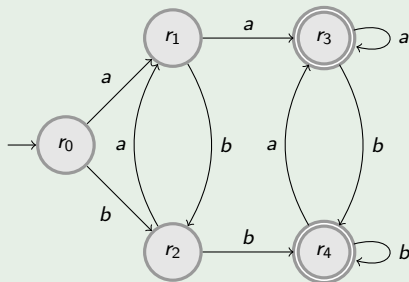


	<i>a</i>	<i>b</i>
$\{q_0\}$	$\{q_2, q_3\}$	$\{q_1, q_2\}$
$\{q_2, q_3\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2\}$
$\{q_1, q_2\}$	$\{q_2, q_3\}$	$\{q_1, q_2, q_4\}$
$\{q_2, q_3, q_4\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2, q_4\}$
$\{q_1, q_2, q_4\}$	$\{q_1, q_2, q_4\}$	$\{q_2, q_3, q_4\}$

Déterminisation

Exemple (Déterminisation)

		<i>a</i>	<i>b</i>
r_0	$\{q_0\}$	$\{q_2, q_3\}$	$\{q_1, q_2\}$
r_1	$\{q_2, q_3\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2\}$
r_2	$\{q_1, q_2\}$	$\{q_2, q_3\}$	$\{q_1, q_2, q_4\}$
r_3	$\{q_2, q_3, q_4\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2, q_4\}$
r_4	$\{q_1, q_2, q_4\}$	$\{q_2, q_3, q_4\}$	$\{q_1, q_2, q_4\}$



C'est tout pour le moment. . .

Des questions ?