THÉORIE DES LANGAGES ET DES AUTOMATES

CH3: LES GRAMMAIRES

Ecole Supérieure Privée
d'Ingénierie et de Technologies

ANNÉE UNIVERSITAIRE 2019 / 2020

PLAN

- Les grammaires
- Langage engendré par une grammaire
- Types de grammaires
- grammaires et dérivation
- Transformation d'une grammaire régulière en un automate fini
- Transformation d’automate fini en une grammaire régulière

Les grammaires

Une grammaire $\mathcal{G}$ est un système de réécriture dans un alphabet $V_n \cup V_t$ formé de deux alphabets disjoints ($V_n \cap V_t = Ø$):

- V_t : l'alphabet terminal, celui des mots engendrés
- V_n : l'alphabet non terminal, ou alphabet des variables, contenant un élément particulier : l'$axiome$, et dont la partie gauche de chaque règle contient au moins une variable.
- $V = V_n \cup V_t$: est appelé le vocabulaire de la grammaire $\mathcal{G}$.

Règle 1

LES GRAMMAIRES

Axiôme

Règle 2

Exemple :

Règle 3

- V_t = {$chat, chien, le, mord, voit$}
- V_n = {$phrase, verbe, groupe\_nominal, article, nom$}

R:

$$
\begin{aligned}
phrase &\rightarrow groupe\_nominal\ verbe\ groupe\_nominal \\
groupe\_nominal &\rightarrow article\ nom \\
article &\rightarrow le \\
nom &\rightarrow chat \\
nom &\rightarrow chien \\
verbe &\rightarrow mord \\
verbe &\rightarrow voit
\end{aligned}
$$

Règle 7

LES GRAMMAIRES

On va pouvoir engendrer ("dériver") huit phrases différentes :

- $phrase \overset{*}{\Rightarrow} le\ chat\ voit\ le\ chien$
- $phrase \overset{*}{\Rightarrow} le\ chat\ voit\ le\ chat$
- $phrase \overset{*}{\Rightarrow} le\ chien\ mord\ le\ chat$
- *etc.*

On a donc besoin de quatre choses différentes pour définir une grammaire :

- V_t : est un ensemble fini de symboles dits terminaux, appelé vocabulaire terminal.
- V_n : est un ensemble fini (disjoint de V_t) de symboles dits non-terminaux, appelé vocabulaire non terminal.
- *L'axiome :* est un non-terminal particulier appelé aussi source.
- *Les règles* : $R \subseteq (V^* \setminus V_t^*) \times V^*$ est un ensemble fini de règles de grammaire notées $g \rightarrow d$ si $(g, d) \in R$.

Les grammaires

Exemple

$$V_t = \{0,1\}$$
$$V_n = \{N, M\}$$
$$Axiôme = N$$

G_1

$$\begin{aligned}
N &\rightarrow 0\\
N &\rightarrow 1\\
N &\rightarrow 1M\\
M &\rightarrow 0\\
M &\rightarrow 1\\
M &\rightarrow 0M\\
M &\rightarrow 1M
\end{aligned}$$

$G2$

$$\begin{aligned}
N &\rightarrow 0\\
N &\rightarrow 1M\\
M &\rightarrow \varepsilon\\
M &\rightarrow 0M\\
M &\rightarrow 1M
\end{aligned}$$

Les deux grammaires décrivent les entiers écrits en binaire

Les grammaires

Exemple

11010

$$G_1 \quad \begin{array}{lll} N & \rightarrow & 0 \\ N & \rightarrow & 1 \\ N & \rightarrow & 1M \\ M & \rightarrow & 0 \\ M & \rightarrow & 1 \\ M & \rightarrow & 0M \\ M & \rightarrow & 1M \end{array}$$

En utilisant la première grammaire :

$$N \rightarrow 1M \rightarrow 11M \rightarrow 110M \rightarrow 1101M \rightarrow 11010$$

$$G_2 \quad \begin{array}{lll} N & \rightarrow & 0 \\ N & \rightarrow & 1M \\ M & \rightarrow & \varepsilon \\ M & \rightarrow & 0M \\ M & \rightarrow & 1M \end{array}$$

En utilisant la deuxième grammaire :

$$N \rightarrow 1M \rightarrow 11M \rightarrow 110M \rightarrow 1101M \rightarrow 11010M \rightarrow 11010$$

Le mot $11010M$ dérive du mot $1M$

$$1\underline{M} \rightarrow 11\underline{M} \rightarrow 110\underline{M} \rightarrow 1101\underline{M} \rightarrow 11010M$$

LES GRAMMAIRES

DÉFINITION

Le langage engendré par la grammaire $\mathcal{G}$ est l'ensemble des mots de V_t qui dérivent de l'axiome de $\mathcal{G}$, que l'on note par $\mathcal{L}(\mathcal{G})$.

REMARQUE

Les grammaires de l'exemple précédent peuvent aussi être spécifiées de la manière suivante :

$$\mathcal{G}_1 \quad \begin{array}{lcl} N & \rightarrow & 0 \mid 1 \mid 1M \\ M & \rightarrow & 0 \mid 1 \mid 0M \mid 1M \end{array}$$

$$\begin{array}{lcl} N & \rightarrow & 0 \mid 1M \\ M & \rightarrow & \varepsilon \mid 0M \mid 1M \end{array} \quad \mathcal{G}2$$

Les types des grammaires

Grammaires de type 0	Grammaires Contextuelles	Grammaires hors-contexte	Grammaires régulières
Aucune restriction sur les règles	Toutes les règles sont de la forme : $g \rightarrow d$ avec $\|g\| \leq \|d\|$.	Toutes les règles sont de la forme : $N \rightarrow u$ avec $u \in V^*$: peut regrouper des terminaux et des non terminaux	Chaque règle peut avoir une des deux formes suivantes : $N \rightarrow u$ ou $N \rightarrow uM$, avec $M, N \in V_n$: des non terminaux et $u \in V_t$: des terminaux Autrement dit : le membre droit possède au plus une occurrence de symbole non terminal, située à l'extrémité droite.

Grammaires contextuelles

La grammaire suivante décrit le langage $\{a^n b^n c^n \mid n \geq 0\}$. Ses règles ont la particularité d'avoir un membre droit de taille supérieure ou égale à celle de leur membre gauche.

$$
\begin{aligned}
S &\rightarrow aSBC \mid \varepsilon \\
CB &\rightarrow BC \\
aB &\rightarrow ab \\
bB &\rightarrow bb \\
bC &\rightarrow bc \\
cC &\rightarrow cc
\end{aligned}
$$

Grammaires contextuelles

$a^3b^3c^3$

$$
\begin{aligned}
S &\rightarrow aSBC \\
&\rightarrow a(aSBC)BC \\
&\rightarrow a(a(aSBC)BC)BC \\
&\rightarrow aaaBCBCBC \\
&\rightarrow aaaBBCCBC \\
&\rightarrow aaaBBCBCC \\
&\rightarrow aaaBBBCCC \\
&\rightarrow aaabBBCCC \\
&\rightarrow aaabbBCCC \\
&\rightarrow aaabbbCCC \\
&\rightarrow aaabbbcCC \\
&\rightarrow aaabbbccC \\
&\rightarrow aaabbbccc
\end{aligned}
$$

$S \rightarrow \varepsilon$

$CB \rightarrow BC$

$CB \rightarrow BC$

$CB \rightarrow BC$

$aB \rightarrow ab$

$bB \rightarrow bb$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

$cC \rightarrow cc$

S – $aSBC \mid \varepsilon$

CB – BC

aB – ab

bB – bb

bC – bc

cC – cc

Grammaires contextuelles

Définition

Étant donnée une grammaire $\mathcal{G} = \{V_t, V_n, S, R\}$, une règle de R est dite contextuelle si elle est de la forme $g \rightarrow d$ avec $|g| \leq |d|$.

Une grammaire est contextuelle si toutes ses règles le sont, et le langage engendré est alors dit contextuel.

Les langages de ce type sont reconnus par des machines de Turing particulières utilisant un espace mémoire borné.

Grammaires hors-contexte

La grammaire suivante décrit le langage $\{a^n b^n \mid n \geq 0\}$. Cette grammaire a une particularité : le membre gauche de chaque règle est un non-terminal..

$$S \rightarrow aSb \mid \varepsilon$$

Grammaires hors-contexte ou algébriques

Définition

Étant donnée une grammaire $\mathcal{G} = \{V_t, V_n, S, R\}$, une règle de R est dite hors-contexte si elle est de la forme $N \rightarrow u$ où $u \in V^*$.

Une grammaire est hors-contexte si toutes ses règles le sont, et le langage engendré est alors dit hors-contexte. On note par LHC la classe des langage hors-contexte.

Les langages hors-contexte sont reconnus par les automates à pile, que nous découvrirons plus loin dans ce cours.

Grammaires régulières

Les grammaires de l'exemple suivant sont hors-contexte, mais elles ont une particularité supplémentaire : **le membre droit possède au plus une occurrence de symbole non terminal, située à l'extrémité droite**.

G_1

$$N \rightarrow 0 \mid 1 \mid 1M$$
$$M \rightarrow 0 \mid 1 \mid 0M \mid 1M$$

$G2$

$$N \rightarrow 0 \mid 1M$$
$$M \rightarrow \varepsilon \mid 0M \mid 1M$$

Grammaires régulières

Définition

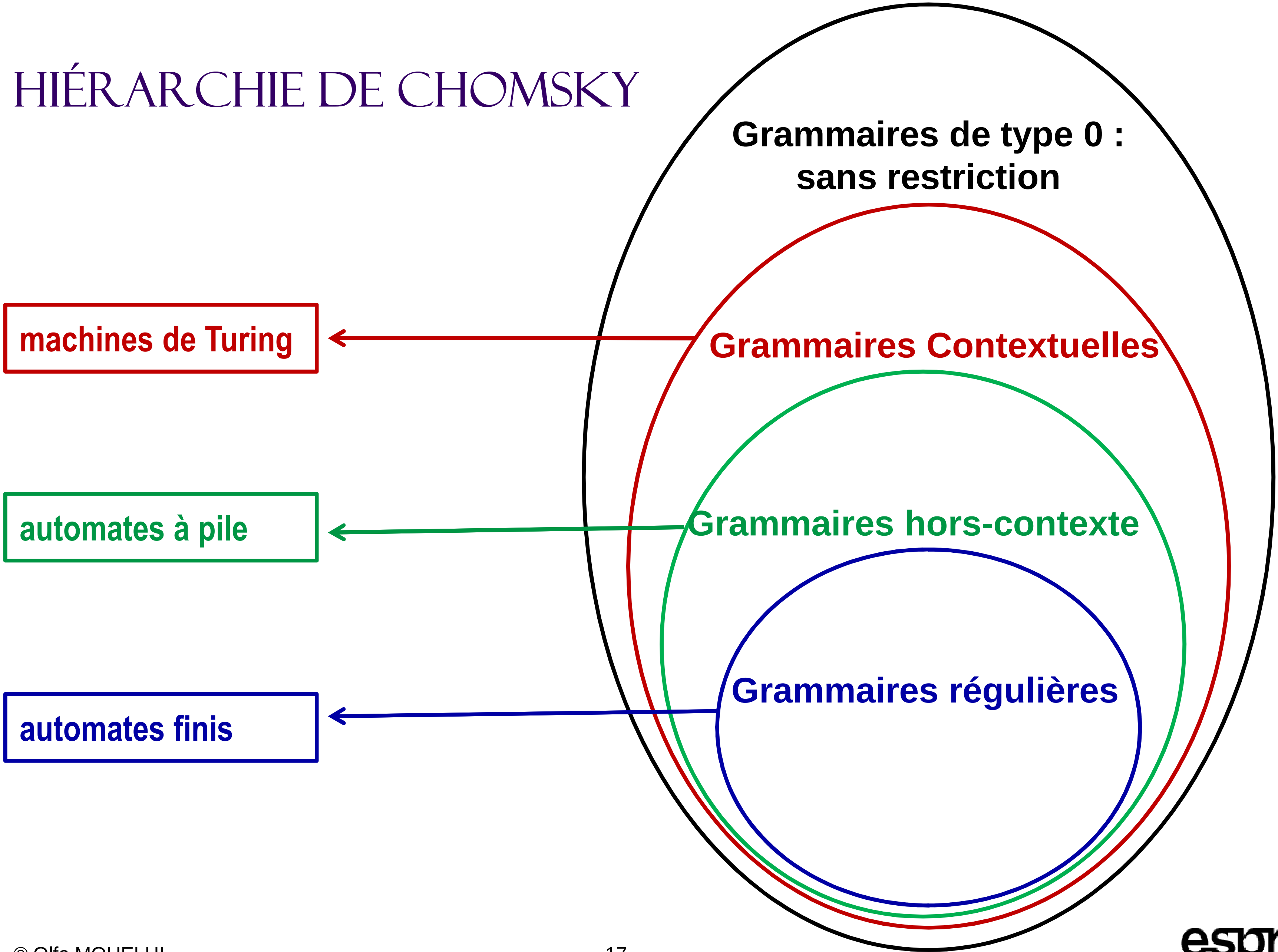
Étant donnée une grammaire $\mathcal{G} = \{V_t, V_n, S, R\}$, une règle de R est dite régulière si elle est de la forme $N \rightarrow u$ ou $N \rightarrow uM$, où $M, N \in V_n$ et $u \in V_t^*$.

Une grammaire est régulière si toutes ses règles le sont, et un langage engendré par une telle grammaire est alors dit régulier ou de type 3.

Les langages réguliers sont bien sûr reconnus par un automate fini.

Un langage est dit régulier si et seulement si il est reconnaissable.

Hiérarchie de Chomsky



Grammaires et dérivation

Considérons le langage des expressions arithmétiques. Cette grammaire aura alors pour symboles terminaux V_t = {+, ×, (,), Id, Cte} et pour non-terminaux l'unique symbole E qui sera donc la source :

$$E \to Id \mid Cte \mid E + E \mid E \times E \mid (E)$$

Cette grammaire permet d'engendrer l'expression arithmétique $x + 2.5 \times 4 + (y + z)$, en supposant que les mots 2.5 et 4 font partie du langage engendré par la grammaire des constantes, et que x, y, z font partie du langage engendré par la grammaire des variables.

Grammaires et dérivation

$$E \rightarrow Id \mid Cte \mid E + E \mid E \times E \mid (E)$$

$$x + 2.5 \times 4 + (y + z)$$

$$E \rightarrow E + E$$
$$E \rightarrow Id + E$$
$$E \rightarrow Id + E \times E$$
$$E \rightarrow Id + Cte \times E$$
$$E \rightarrow Id + Cte \times E + E$$
$$E \rightarrow Id + Cte \times Cte + E$$
$$E \rightarrow Id + Cte \times Cte + (E)$$
$$E \rightarrow Id + Cte \times Cte + (E + E)$$
$$E \rightarrow Id + Cte \times Cte + (Id + E)$$
$$E \rightarrow Id + Cte \times Cte + (Id + Id)$$

$$x \quad 2.5 \quad 4 \quad y \quad z$$

Chaque fois on prend le non terminal le plus à gauche $\Rightarrow$ **dérivation à gauche.**

Grammaires et dérivation

$E \to Id \mid Cte \mid E + E \mid E \times E \mid (E)$

$x + 2.5 \times 4 + (y + z)$

$E \to E + E$

$E \to E + (E)$

$E \to E + (E + E)$

$E \to E + (E + Id)$

$E \to E + (Id + Id)$

$E \to E \times E + (Id + Id)$

$E \to E \times Cte + (Id + Id)$

$E \to E + E \times Cte + (Id + Id)$

$E \to E + Cte \times Cte + (Id + Id)$

$E \to Id + Cte \times Cte + (Id + Id)$

$\quad x \quad 2.5 \quad 4 \quad y \quad z$

Chaque fois on prend le non terminal le plus à droite $\Rightarrow$ **dérivation à droite**.

Grammaires et dérivation

- Étant donné la grammaire des expressions arithmétiques :
  $E \rightarrow Id \mid Cte \mid E + E \mid E \times E \mid (E)$
- Considérons la phrase $Id + Id \times Id$.

Dérivation à droite

$$\begin{aligned} E &\rightarrow E + E \\ &\rightarrow Id + E \\ &\rightarrow Id + E \times \mathrm{E} \\ &\rightarrow Id + Id \times \mathrm{E} \\ &\rightarrow Id + Id \times Id \end{aligned}$$

Dérivation à gauche

$$\begin{aligned} E &\rightarrow \mathrm{E} \times \mathrm{E} \\ &\rightarrow E + E \times E \\ &\rightarrow Id + E \times E \\ &\rightarrow Id + Id \times E \\ &\rightarrow Id + Id \times Id \end{aligned}$$

DÉFINITION

Une grammaire $\mathcal{G}$ est **ambiguë** s'il existe un mot $w \in \mathcal{L}(\mathcal{G})$ qui possède plus qu'une dérivation gauche (droite) dans $\mathcal{G}$.

Exemple:

$E \rightarrow E+E|E*E|\ (E)|\ x1|\ \ x2$

2 dérivations à gauche pour w

Soit w = x1+x1*x2

Cette grammaire est ambiguë car la chaîne w a plus d'une dérivation gauche:

$E \rightarrow \underline{E}+E \rightarrow x1+\underline{E} \rightarrow x1+\underline{E}*E \rightarrow x1+x1*\underline{E} \rightarrow x1+x1*x2$

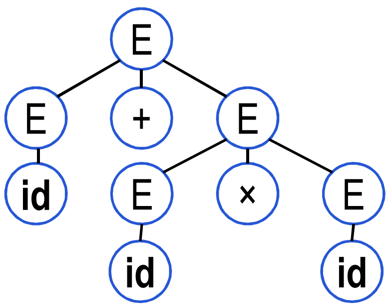
$E \rightarrow \underline{E}*E \rightarrow \underline{E}+E*E \rightarrow x1+\underline{E}*E \rightarrow x1+x1*\underline{E} \rightarrow x1+x1*x2$

De même on peut trouver deux dérivations droites

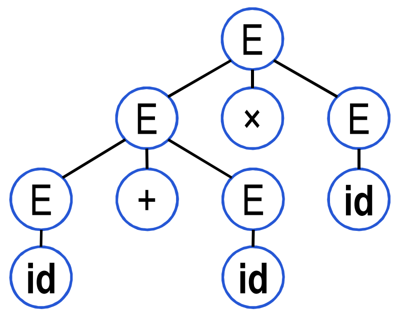
Représentation arborescente

Exemple : id + id × id

$$\begin{aligned} E &\Rightarrow E + E \\ &\Rightarrow \mathrm{id} + E \\ &\Rightarrow \mathrm{id} + E \times E \\ &\Rightarrow \mathrm{id} + \mathrm{id} \times E \\ &\Rightarrow \mathrm{id} + \mathrm{id} \times \mathrm{id} \end{aligned}$$



$$\begin{aligned} E &\Rightarrow E \times E \\ &\Rightarrow E + E \times E \\ &\Rightarrow \mathrm{id} + E \times E \\ &\Rightarrow \mathrm{id} + \mathrm{id} \times E \\ &\Rightarrow \mathrm{id} + \mathrm{id} \times \mathrm{id} \end{aligned}$$



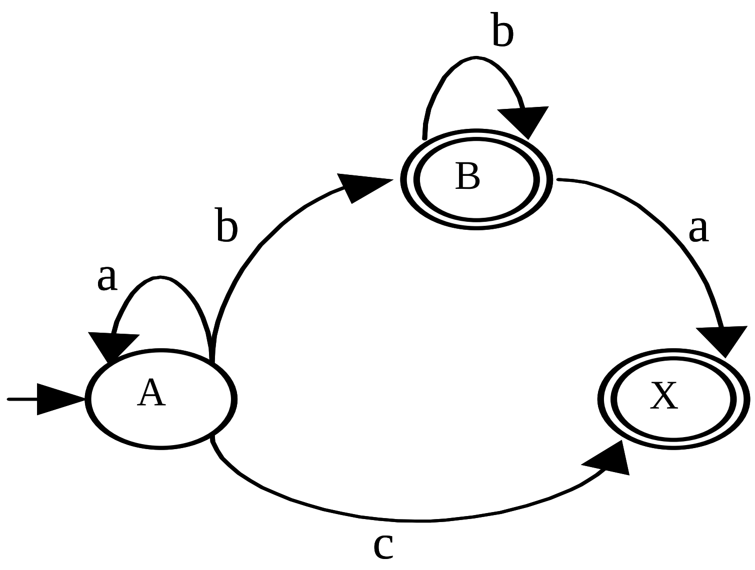
Transformation d'une grammaire régulière en un automate fini :

Soit G=(V,T,S,R) une grammaire régulières, on peut construire un automate fini A=(Q,Σ,δ,q_0,F) qui accepte les mots engendrés par G:

- Pour toutes les productions $B\rightarrow\alpha$ de P, on ajoute un nouveau symbole non terminal X et on remplace ces productions par $B\rightarrow\alpha X$ et $X\rightarrow\varepsilon$
- L'ensemble des états Q de l'automate est celui des non-terminaux. Q=V
- L'alphabet Σ est égal à l'ensemble T des terminaux. Σ=T
- L'état q_0 correspond à l'axiome A de la grammaire. q_0 =S
- δ= {(p,x,q)/ il existe une règle de la forme $p\rightarrow xq$}
- F={X/ $X\rightarrow\varepsilon$}

Exemple

- G=(V,T,A,R) une grammaire régulière
- V={A,B} axiome A
- T={a,b,c}
- R={A→aA
  A→bB
  A→c **(devient A→cX et X→ε)**
  B→bB
  B→a **(devient B→aX)**
  B→ε



TRANSFORMATION D'UN AUTOMATE FINI EN UNE GRAMMAIRE RÉGULIÈRE:

Soit un automate fini donné $M=(Q,\Sigma,\delta, q_0,F)$

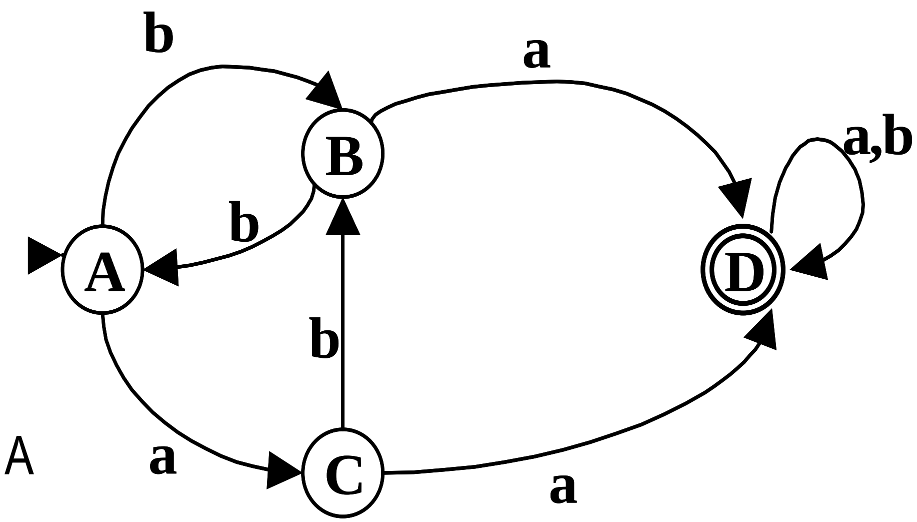
La grammaire $G=(V,T,S,R)$ équivalente est définie comme suit :

- $V=Q$
- $T= \Sigma$
- $S= q_0$
- $R=\{p\longrightarrow xq/(p,x,q) \in \delta \}\cup\{Q\longrightarrow \varepsilon /Q\in F\}$

Exemple :

- A=(Q,Σ,δ, q0=A,F)

  Σ={a,b},Q={A,B,C,D},F={D}



- G=(V,T,A,R)

  V ={A,B,C,D},T={a,b}, axiome A

  R={A→aC, A→bB, B→aD, B→bA, C→aD, C→bB, D→aD, D→bD, D→ε}