3

Les grammaires et les langages algébriques.

1 – Définitions de base.

Une grammaire est une machine à réécrire c'est–à–dire, une machine à faire des substitutions.

Les grammaires en général.

Une grammaire est un triplet $G = (\mathcal{V}, \mathcal{A}, R)$ où :

- $\mathcal{V}$ est l'alphabet **fini** des non–terminaux ou **variables**,
- $\mathcal{A}$ est l'alphabet **fini**, des terminaux ou **constantes**, $\mathcal{A}$ et $\mathcal{V}$ **sont supposés disjoints** ;
- R est un ensemble **fini** de couples $(\lambda, \alpha) \in (\mathcal{A} + \mathcal{V})^* \times (\mathcal{A} + \mathcal{V})^*$ appelés **règles** de production de G.

  Une règle (λ, α) sera représentée de façon plus suggestive par

  $$\lambda \underset{G}{\longrightarrow} \alpha$$

  plus simplement $\lambda \longrightarrow \alpha$.

Fonctionnement d'une grammaire.

• Une règle $\lambda \underset{G}{\longrightarrow} \alpha$ agit sur un mot de la forme $\beta_1\lambda\beta_2$ en remplaçant λ par α, ce que l'on schématise par

$$\beta_1\lambda\beta_2 \underset{G}{\Longrightarrow} \beta_1\alpha\beta_2.$$

• *Une dérivation* (ou *inférence*) est un enchaînement fini de telles opérations.

• L'ensemble des mots de $\mathcal{A}^*$ que l'on peut inférer à partir d'un mot donné μ est *le langage engendré par G à partir de μ.*

1.1 – Grammaires indépendantes du contexte.

On se limitera aux grammaires dites *indépendantes du contexte* (context–free grammars) ou *algébriques*.

Grammaires indépendantes du contexte

Une grammaire est dite *indépendante du contexte* lorsque leurs règles ont la forme

$$X \underset{G}{\longrightarrow} \alpha$$

avec $X \in \mathcal{V}$ et $\alpha \in (\mathcal{A} + \mathcal{V})^*$.

Nous les appellerons simplement **grammaires**.

La définition présente les règles de façon individuelle. Il est commode de regrouper les règles relatives à une même variable.

Règles globales

La *règle globale* $X \longrightarrow \mathbf{l}(X)$ de X dans G est définie par

$$\alpha \in \mathbf{l}(X) \text{ ssi } X \longrightarrow \alpha \text{ est une règle de } G$$

pour tout $\alpha \in (\mathcal{A} + \mathcal{V})^*$.

Exemple 1.

Soit $G = (\mathcal{V}, a + b, R)$ où $\mathcal{V} = S$. On peut décrire R

- ou bien comme un ensemble de règles individuelles, par exemple :

$$\begin{aligned} S &\longrightarrow SbS \\ S &\longrightarrow \varepsilon \\ S &\longrightarrow a \\ S &\longrightarrow aa \end{aligned}$$

- ou bien par la règle globale correspondante :

$$S \longrightarrow SbS + \varepsilon + a + aa$$

Commentaires.

$\mathbf{l}(X)$ est donc l'ensemble de tous les seconds membres des règles pour X : c'est un langage **fini**, puisque R l'est par définition.

On peut décrire directement une grammaire comme un triplet $G = (\mathcal{V}, \mathcal{A}, \mathbf{l})$ où

$$\mathbf{l} : \mathcal{V} \to \mathcal{P}((\mathcal{A} + \mathcal{V})^*)$$

est une substitution finie.

1.2 – Dérivations dans une grammaire.

Les dérivations dans $G = (\mathcal{V}, \mathcal{A}, R)$ sont des suites de réécritures que l'on peut opérer sur les éléments de $(\mathcal{A}+\mathcal{V})^*$ par application de ses règles.

• *Une dérivation élémentaire* est l'application d'une règle de la grammaire.

Une dérivation élémentaire se définit pour $\beta_1 \in (\mathcal{A}+\mathcal{V})^*$, $\beta_2 \in (\mathcal{A}+\mathcal{V})^*$, $X \in \mathcal{V}$ par

$$\beta_1 X \beta_2 \underset{G}{\overset{1}{\Longrightarrow}} \beta_1 \alpha \beta_2$$

pour toute règle $\varrho : X \underset{G}{\longrightarrow} \alpha$.

C'est la *mise de la règle ϱ dans le contexte* (β_1, β_2).

• *Une dérivation de longueur i*, notée $\beta \underset{G}{\overset{i}{\Longrightarrow}} \gamma$ est une suite de i dérivations élémentaires qui s'enchaînent, qui se définit par récurrence sur i :

$$0 \quad : \beta \underset{G}{\overset{0}{\Longrightarrow}} \beta \qquad \text{(pas d'application de règle)}$$

$$i \mapsto i+1 \quad : \beta \underset{G}{\overset{i}{\Longrightarrow}} \delta \underset{G}{\overset{1}{\Longrightarrow}} \gamma \qquad \text{(une application de plus)}$$

• On notera $\beta \underset{G}{\overset{*}{\Longrightarrow}} \gamma$ pour signifier

"*il existe une dérivation* $\beta \underset{G}{\overset{i}{\Longrightarrow}} \gamma$".

Exemple 1 (suite).

Reprenons $G = (\mathcal{V}, a + b, R)$ où $\mathcal{V} = S$ et R est défini la règle globale $S \longrightarrow SbS + \varepsilon + a + aa$.

Voici une dérivation $S \overset{11}{\Longrightarrow} aabbabaabba$:

$$
\begin{array}{lrll}
(1) & \cdot S & \overset{1}{\Longrightarrow} \cdot SbS & (S \longrightarrow SbS) \\
(2) & & \overset{1}{\Longrightarrow} SbSb \cdot S & (S \longrightarrow SbS) \\
(3) & & \overset{1}{\Longrightarrow} \cdot SbSbSbS & (S \longrightarrow SbS) \\
(4) & & \overset{1}{\Longrightarrow} SbSb \cdot SbSbS & (S \longrightarrow SbS) \\
(5) & & \overset{1}{\Longrightarrow} SbSbSbSb \cdot SbS & (S \longrightarrow SbS) \\
(6) & & \overset{1}{\Longrightarrow} SbSbSbSbb \cdot S & (S \longrightarrow \varepsilon) \\
(7) & & \overset{1}{\Longrightarrow} \cdot SbSbSbSbba & (S \longrightarrow a) \\
(8) & & \overset{1}{\Longrightarrow} aab \cdot SbSbSbba & (S \longrightarrow aa) \\
(9) & & \overset{1}{\Longrightarrow} aabb \cdot SbSbba & (S \longrightarrow \varepsilon) \\
(10) & & \overset{1}{\Longrightarrow} aabbab \cdot Sbba & (S \longrightarrow a) \\
(11) & & \overset{1}{\Longrightarrow} aabbabaabba & (S \longrightarrow aa)
\end{array}
$$

L'occurrence de la variable à laquelle on applique une règle est précédée d'un point (que la tradition nomme *pointeur*).

Langages algébriques

Le *langage engendré par une grammaire* $G = (\mathcal{V}, \mathcal{A}, R)$ *à partir de* $\alpha \in (\mathcal{A}+\mathcal{V})^*$ est $\mathcal{L}(G,\alpha) \subseteq \mathcal{A}^*$ défini par

$$u \in \mathcal{L}(G,\alpha) \text{ ssi } \alpha \overset{*}{\underset{G}{\Longrightarrow}} u$$

pour tout $u \in \mathcal{A}^*$.

Un langage $L \subseteq \mathcal{A}^*$ est dit *algébrique* ssi il existe une grammaire $G = (\mathcal{V}, \mathcal{A}, R)$ et $S \in \mathcal{V}$ telles que

$$L = \mathcal{L}(G,S).$$

On considère aussi *le langage étendu engendré par* G *à partir de* $\alpha \in (\mathcal{A}+\mathcal{V})^*$, $\hat{\mathcal{L}}(G,\alpha) \subseteq (\mathcal{A}+\mathcal{V})^*$ défini par

$$\beta \in \hat{\mathcal{L}}(G,\alpha) \text{ ssi } \alpha \overset{*}{\underset{G}{\Longrightarrow}} \beta.$$

On a évidemment $\mathcal{L}(G,\alpha) = \hat{\mathcal{L}}(G,\alpha) \cap \mathcal{A}^*$.

La seule façon de prouver qu'un langage est algébrique est de montrer l'existence d'une grammaire qui engendre ce langage à partir de l'une de ses variables.

1.3 – Les grammaires linéaires.

Une grammaire $G = (\mathcal{V}, \mathcal{A}, R)$ est dite *linéaire* lorsque dans chacune de ses règles $X \longrightarrow \alpha$, le mot $\alpha \in (\mathcal{A} + \mathcal{V})^*$ comporte **au plus une occurrence de variable**. Elles ont donc l'une des formes

$$X \longrightarrow uYv \text{ ou } X \longrightarrow u$$

pour $X \in \mathcal{V}$, $Y \in \mathcal{V}$, $u \in \mathcal{A}^*$ et $v \in \mathcal{A}^*$.

Cette propriété s'étend aux dérivations : *pour toute variable X et toute dérivation $X \stackrel{k}{\Longrightarrow} \alpha$ dans une grammaire linéaire, α comporte au plus une occurrence de variable.*

Exemple.

$G = (\mathcal{V}, \mathcal{A}, R)$ pour laquelle $\mathcal{V} = S + X + Y$, $\mathcal{A} = a + b + c$ et dont les règles glogales sont

$$\begin{aligned} S &\longrightarrow aXa + bYb + cSc \\ X &\longrightarrow cY + Sa + b \\ Y &\longrightarrow cX + Sb + a \end{aligned}$$

est linéaire.

Voici une dérivation dans G :

$$\begin{aligned} S &\stackrel{1}{\Longrightarrow} cSc && (S \longrightarrow cSc) \\ &\stackrel{1}{\Longrightarrow} caXac && (S \longrightarrow aXa) \\ &\stackrel{1}{\Longrightarrow} caSaac && (X \longrightarrow Sa) \\ &\stackrel{1}{\Longrightarrow} cabYbaac && (S \longrightarrow bYb) \\ &\stackrel{1}{\Longrightarrow} cababaac && (Y \longrightarrow a) \end{aligned}$$

Grammaires linéaires à droite.

On qualifie ainsi les grammaires linéaires dont chaque règle est de l'une des formes

$$X \longrightarrow xY \text{ ou } X \longrightarrow \varepsilon$$

pour $x \in \mathcal{A}$ et $Y \in \mathcal{V}$.

Propriété

Tout langage régulier est algébrique.

Plus précisément :

Un langage engendré par une grammaire linéaire à droite est régulier, et réciproquement, tout langage régulier est engendré par une grammaire linéaire à droite appropriée.

Les constructions se font en appliquant les correspondances présentées dans le tableau ci–desous.

Grammaire	AF
$\mathcal{V}$	Q
Axiome	Entrée
$X \to xY$	$Y \in X \bullet x$
$X \to \varepsilon$	$X \in F$

☞ Soient $G = (\mathcal{V}, \mathcal{A}, R)$ une grammaire linéaire à droite et $S \in \mathcal{V}$: on va construire un AF $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ à une seule entrée, tel que $\mathcal{L}(\mathbf{A}) = \mathcal{L}(G, S)$:

- $Q = \mathcal{V}$: on notera q_X l'état correspondant à la variable X ;
- $q_0 = q_S$;
- pour toute $X \in \mathcal{V}$, toute $Y \in \mathcal{V}$ et tout $x \in \mathcal{A}$: $q_Y \in \delta(q_X, x)$ ssi $(X \longrightarrow xY)$ est une règle de G ;
- $q_X \in F$ ssi $X \longrightarrow \varepsilon$ est une règle de G.

☞ Soit $\mathbf{A} = (Q, \mathcal{A}, \bullet, q_0, F)$ un AF : on va construire une grammaire linéaire à droite $G = (\mathcal{V}, \mathcal{A}, R)$ et trouver $S \in \mathcal{V}$ tels que $\mathcal{L}(G, S) = \mathcal{L}(\mathbf{A})$:

- $\mathcal{V} = Q$: on notera X_q la variable correspondant à l'état q ;
- $S = X_{q_0}$;
- pour tout $q \in Q$, tout $r \in Q$ et tout $x \in \mathcal{A}$: $X_q \longrightarrow xX_r$ est une règle de G ssi $q \bullet x = r$;
- pour tout $q \in Q$: $X_q \longrightarrow \varepsilon$ est une règle de G ssi $q \in F$.

2 – Représentation arborescente.

Une dérivation est l'enchaînement séquentiel de dérivations élémentaires, mais ceci est généralement obtenu au prix de choix qui, en réalité, sont arbitraires.

Soient $Y \longrightarrow \beta$ et $Y' \longrightarrow \beta'$ deux règles de G. Elles sont applicables à tout mot de la forme $\alpha_1 Y \alpha_2 Y' \alpha_3$ pour produire les deux dérivations :

$$d_1 : \alpha_1 Y \alpha_2 Y' \alpha_3 \overset{1}{\Longrightarrow} \alpha_1 \beta \alpha_2 Y' \alpha_3 \overset{1}{\Longrightarrow} \alpha_1 \beta \alpha_2 \beta' \alpha_3,$$
$$d_2 : \alpha_1 Y \alpha_2 Y' \alpha_3 \overset{1}{\Longrightarrow} \alpha_1 Y \alpha_2 \beta' \alpha_3 \overset{1}{\Longrightarrow} \alpha_1 \beta \alpha_2 \beta' \alpha_3.$$

La commutation de règles dans une dérivation est le remplacement de d_1 par d_2 ou celui de d_2 par d_1 : une telle opération ne modifie pas la nature d'une dérivation mais seulement la façon de la décrire.

Dérivations équivalentes

Deux dérivations sont dites *équivalentes* ssi on peut transformer l'une en l'autre par une suite de commutations de règles.

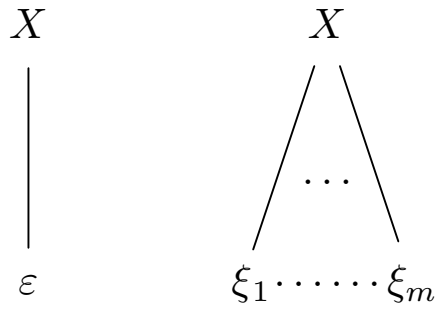
La représentation arborescente caractérise l'équivalence entre dérivations de façon plus satisfaisante.

2.1 – Arbres de dérivation d'une grammaire.

L'idée est simplement d'arborer chaque règle

$$X \longrightarrow \varepsilon \qquad\qquad X \longrightarrow \xi_1 \dots \xi_m$$

de G, en la représentant sous la forme d'un embranchement ordonné, sur le modèle ci–dessous



et de construire des arbres au lieu de dérivations.

Définition des arbres de dérivation.

Dans un arbre

- un nœud est étiqueté par un élément de $\mathcal{V}$ et une feuille par un élément de $\mathcal{A} + \mathcal{V}$,
- l'embranchement d'un nœud étiqueté par $X \in \mathcal{V}$ est déterminé par une règle $X \longrightarrow \alpha$ de G.

Les descendants immédiats d'un tel nœud sont étiquetés par les caractères successifs de α.

Dans le cas où $\alpha = \varepsilon$, ε n'est pas une feuille, mais l'indication de **l'absence définitive de toute feuille**!

$\mathcal{D}(G)$ désigne l'ensemble des arbres de dérivation de G.

Un arbre $\mathbf{a} \in \mathcal{D}(G)$ est désigné schématiquement par :

$$\mathbf{a} : \xi \overset{i}{\underset{G}{\Longrightarrow}} \gamma$$

où

- $\xi \in \mathcal{A} + \mathcal{V}$ est l'étiquette de la *racine*,
- $\gamma \in (\mathcal{A} + \mathcal{V})^*$ est la *frondaison*, c'est–à–dire, le mot constitué des étiquettes de ses feuilles,
- i est le nombre de ses nœuds.

Lorsque G est fixée,on écrit simplement $\mathbf{a} : \xi \overset{i}{\Longrightarrow} \gamma$.

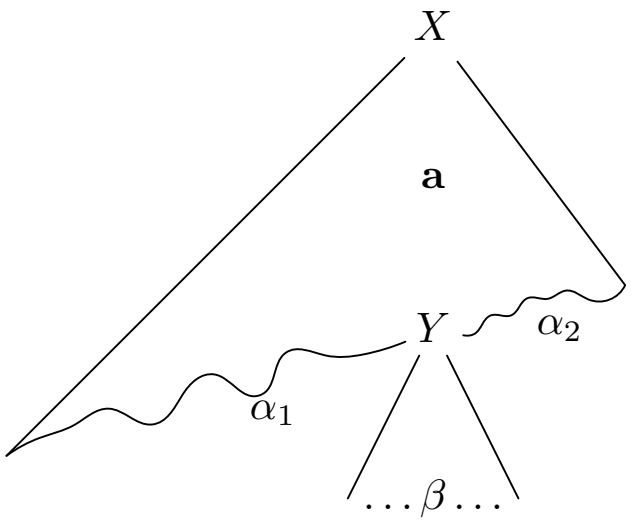
Construction inductive de $\mathcal{D}(G)$.

Elle utilise deux types particuliers d'arbres :

- *Les arbres triviaux* : tout $\xi \in \mathcal{A} + \mathcal{V}$ définit un arbre $\xi \overset{0}{\Longmapsto} \xi$ dont la racine est une feuille étiquetée ξ.
- *Les arbres élémentaires* : une règle $X \longrightarrow \alpha$ définit un arbre $X \overset{1}{\Longmapsto} \alpha$, sur le modèle de la figure 1.

et une opération :

- *La greffe* d'un arbre élémentaire :
  soit un arbre $\mathbf{a} : X \overset{i}{\Longmapsto} \alpha_1 Y \alpha_2$ et soit $\mathbf{b} : Y \overset{1}{\Longmapsto} \beta$ un arbre élémentaire défini par une règle $Y \longrightarrow \beta$.
  On obtient un arbre $\mathbf{c} : X \overset{i+1}{\Longmapsto} \alpha_1 \beta \alpha_2$ en greffant $\mathbf{b}$ sur la feuille Y de $\mathbf{a}$:



2.2 – Arbre d'une dérivation $\xi \overset{i}{\Longrightarrow} \gamma$.

Pour tout $\xi \in \mathcal{A}+\mathcal{V}$, tout $\gamma \in (\mathcal{A}+\mathcal{V})^*$ et toute dérivation $d : \xi \overset{i}{\Longrightarrow} \gamma$, on construit l'*arbre de d* qui est noté

$$\mathbf{arb}(d) : \xi \overset{i}{\Longmapsto} \gamma$$

dont le nombre de nœuds i est égal à la longueur de d.

La construction se fait par récurrence sur i :

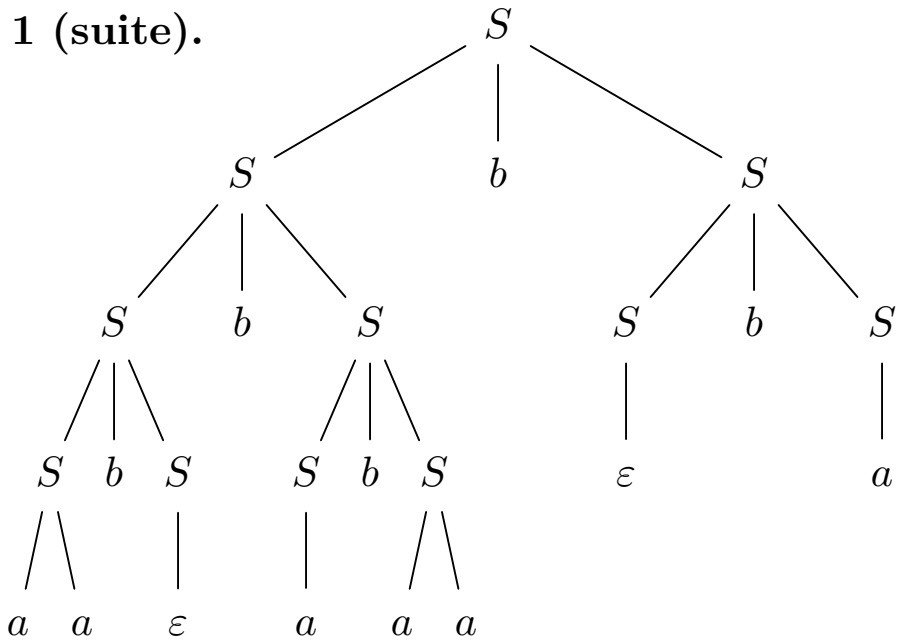
0 : si $d : \xi \overset{0}{\Longrightarrow} \gamma$ alors $\gamma = \xi$ et $\mathbf{arb}(d)$ est trivial.

Dans les autres cas, ξ est une variable X.

$i \mapsto i+1$: si $d : X \overset{i}{\Longrightarrow} \gamma_1 Y \gamma_2 \overset{1}{\Longrightarrow} \gamma_1 \beta \gamma_2$ est la composée de $d' : X \overset{i}{\Longrightarrow} \gamma_1 Y \gamma_2$ et d'une dérivation élémentaire définie par la règles $Y \longrightarrow \beta$, $\mathbf{arb}(d)$ s'obtient en greffant l'arbre élémentaire représentant cette règle sur la feuille de $\mathbf{arb}(d')$ étiquetée par l'occurrence en cause de Y.

Exemple 1 (suite).

La figure :



représente l'arbre de la dérivation :

$$
\begin{aligned}
\cdot S &\overset{1}{\Longrightarrow} \cdot SbS && (S \longrightarrow SbS)\\
&\overset{1}{\Longrightarrow} \cdot SbSbS && (S \longrightarrow SbS)\\
&\overset{1}{\Longrightarrow} \cdot SbSbSbS && (S \longrightarrow SbS)\\
&\overset{1}{\Longrightarrow} aab \cdot SbSbS && (S \longrightarrow aa)\\
&\overset{1}{\Longrightarrow} aabb \cdot SbS && (S \longrightarrow \varepsilon)\\
&\overset{1}{\Longrightarrow} aabb \cdot SbSbS && (S \longrightarrow SbS)\\
&\overset{1}{\Longrightarrow} aabbab \cdot SbS && (S \longrightarrow a)\\
&\overset{1}{\Longrightarrow} aabbabaab \cdot S && (S \longrightarrow aa)\\
&\overset{1}{\Longrightarrow} aabbabaab \cdot SbS && (S \longrightarrow SbS)\\
&\overset{1}{\Longrightarrow} aabbabaabb \cdot S && (S \longrightarrow \varepsilon)\\
&\overset{1}{\Longrightarrow} aabbabaabba && (S \longrightarrow a)
\end{aligned}
$$

2.3 – Arborescence d'une dérivation $\beta \stackrel{i}{\Longrightarrow} \gamma$.

Une construction inductive de l'ensemble des arborescences de dérivation de G s'obtient en adaptant celle des arbres.

Nous désignerons une arbrorescence par

$$\mathbf{u} : \beta \stackrel{i}{\Longmapsto} \gamma$$

où $\beta \in (\mathcal{A} + \mathcal{V})^*$ est le mot constitué des étiquettes de ses racines, $\gamma \in (\mathcal{A} + \mathcal{V})^*$ sa frondaison et i le nombre de ses nœuds.

En plus des arbres élémentaires, la construction utilise un type particulier d'arborescence :

- *Les arbrorescences triviales* : tout $\beta \in (\mathcal{A} + \mathcal{V})^*$ définit une arborescence $\beta \stackrel{0}{\Longmapsto} \beta$ (en particulier $\varepsilon \stackrel{0}{\Longmapsto} \varepsilon$), constituée de feuilles étiquetées par les caractères successifs de β.

et l'opération de greffe d'arbres élémentaires :

la frondaison d'une arborescence a exactement le même aspect que celle d'un arbre !

La construction de l'arborescence

$$\mathbf{arb}(d) : \beta \stackrel{i}{\Longmapsto} \gamma$$

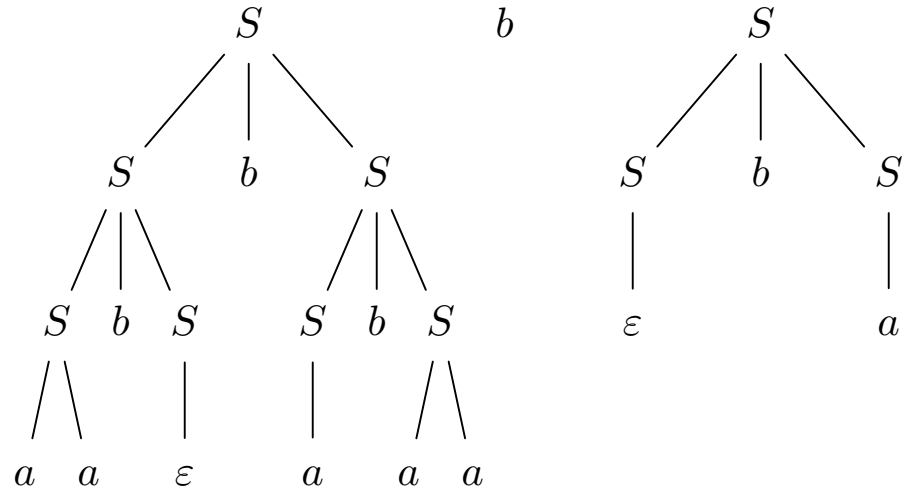
d'une dérivation $d : \beta \stackrel{i}{\Longrightarrow} \gamma$ est identique à celle qui a été faite plus haut pour les arbres, mais en partant cette fois de l'arborescence triviale définie par β.

Exemple 1 (suite).

Si, dans la dérivation de l'exemple 1, on efface la première dérivation élémentaire, on obtient une dérivation

$$SbS \overset{10}{\Longrightarrow} aabbabaabba$$

dont l'arborescence est représentée par la figure :



La construction des arborescences est celle de juxtapositions d'arbres :

- initialement, tous ces arbres sont triviaux,
- ensuite, chaque greffe se produit sur l'un des arbres de l'arborescence déjà construite.

Une arborescence de dérivation de G est un mot sur $\mathcal{D}(G)$.

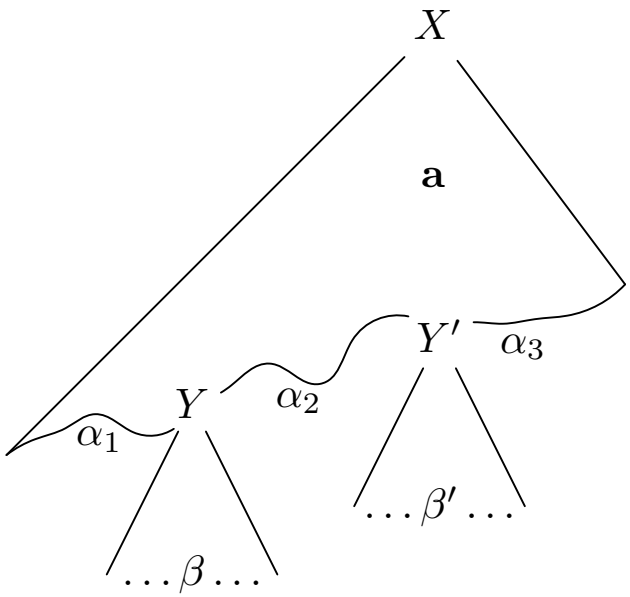
Dérivations équivalentes : suite.

Nous pouvons maintenant caractériser l'équivalence des dérivations de façon géométrique :

Equivalence de dérivations

Deux dérivations sont équivalentes ssi elles ont la même arborescence.

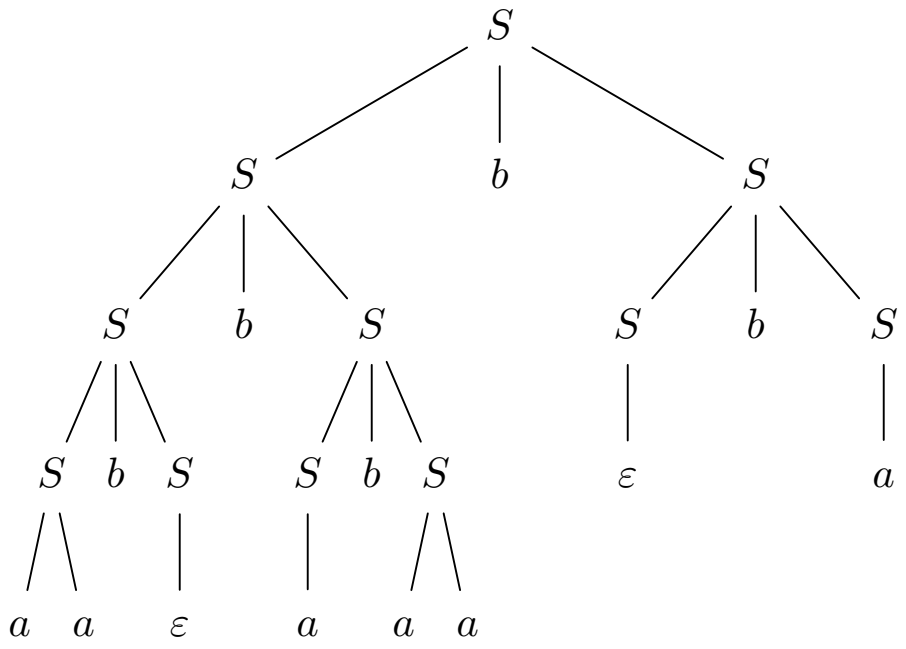
Il suffit essentiellement de vérifier que la commutation de règles ne modifie pas l'arborescence d'une dérivation. En reprenant les notations du début, la figure



illustre ce fait : les deux greffes se produisant en deux points différents de la même frondaison, l'ordre dans lequel on les effectue n'influence pas le résultat.

Exemple.

L'arbre



est celui des deux dérivations des suites de l'exemple 1 :

$$S \overset{11}{\Longrightarrow} aabbabaabba$$

données plus haut.

On peut effectivement vérifier qu'elles sont équivalentes.

2.4 – Séquentialisation d'une arborescence.

La *séquentialisation* d'une arborescence $\mathbf{u} : \beta \overset{i}{\Longmapsto} \gamma$ est la construction d'une dérivation $d : \beta \overset{i}{\Longrightarrow} \gamma$ pour laquelle $\mathbf{arb}(d) = \mathbf{u}$.

Méthode ascendante.

La construction de d se fait par récurrence sur i : la méthode utilisée est dite *ascendante* car la dérivation est construite en partant de la fin.

- si $\mathbf{u}$ est triviale alors $\gamma = \beta$ et $d : \beta \overset{0}{\Longrightarrow} \beta$ est une dérivation triviale,
- supposons la construction possible pour toute arborescence à i nœuds et soit $\mathbf{u} : \beta \overset{i+1}{\Longmapsto} \gamma$ a $i+1$ nœuds ; choisissons un nœud de $\mathbf{u}$ extrémal :
  - ce nœud et ses embranchements forment l'arbre élémentaire d'une règle $Y \longrightarrow \alpha$,
  - l'arborescence $\mathbf{u}'$ obtenu en "élagant" les embranchements du nœud choisi, a la forme $\beta \overset{i}{\Longmapsto} \gamma_1 Y \gamma_2$ où l'étiquette Y est celle du nœud de $\mathbf{u}$ qui vient de devenir une feuille de $\mathbf{u}'$, par l'HR, c'est l'arborescence d'une dérivation $d' : \beta \overset{i}{\Longrightarrow} \gamma_1 Y \gamma_2$.

  Il reste à appliquer une dérivation élémentaire définie avec la règle $Y \longrightarrow \alpha$, pour obtenir une dérivation

$$d : \beta \overset{i}{\Longrightarrow} \gamma_1 Y \gamma_2 \overset{1}{\Longrightarrow} \gamma_1 \alpha \gamma_2$$

  dont $\mathbf{u}$ est l'arborescence.

La propriété $\mathbf{arb}(d) = \mathbf{u}$ tient au fait que l'élagage utilisé dans la construction de d est l'opération réciproque de la greffe d'un arbre élémentaire utilisée dans la construction de $\mathbf{arb}(d)$.

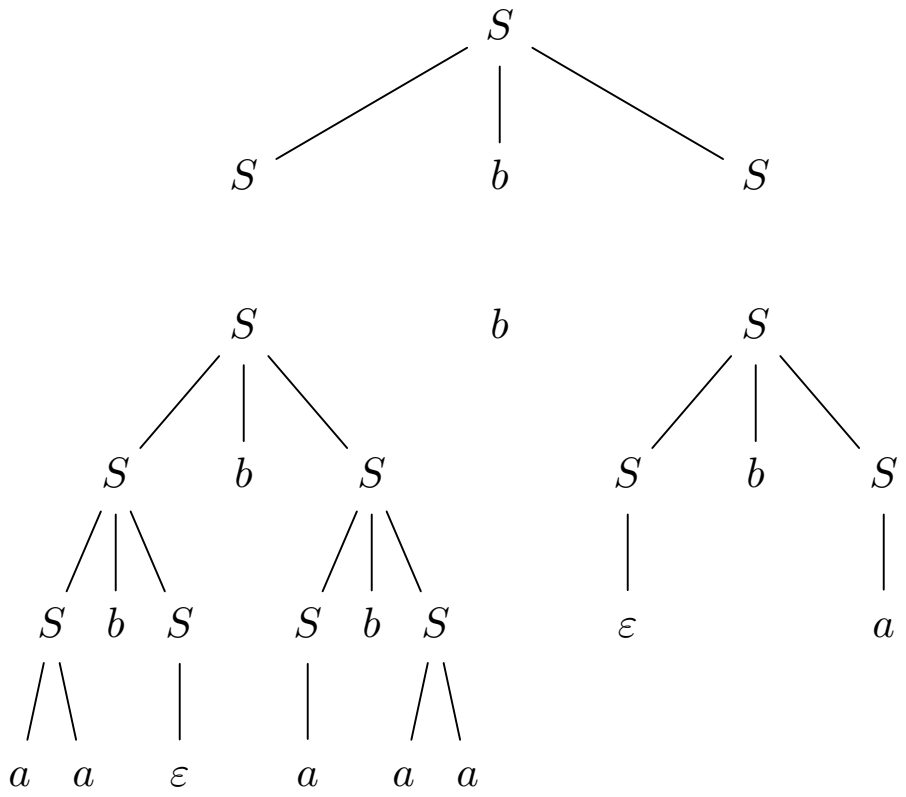
Remarques.

- En général, $\mathbf{u}$ admet plusieurs nœuds extrémaux : un choix est donc possible à chaque étape de la construction de d. Tous ces choix produisent des dérivations équivalentes.
- Si $\mathbf{u} = \mathbf{arb}(d)$, il est facile de vérifier que d est une des séquentialisations de $\mathbf{u}$: si on numérote les nœuds dans l'ordre où ils se présentent dans la construction de $\mathbf{arb}(d)$, il suffit, à chaque étape de la séquentialisation, d'élaguer le nœud qui porte le numéro le plus grand.

Méthode descendante.

La dérivation est construite en partant de son début : elle introduit des arborescences même quand on veut séquentialiser un arbre.

Dans l'étape de la récurrence, on élague un embranchement issu de l'une des racines. La figure montre ce qui se passe lorsque l'on applique ce principe à l'arbre précédent : la partie restante est une arborescence.



3 – Le lemme principal.

Résumons : *une arborescence est une façon de représenter une dérivation.* Dans la pratique, on parle de dérivations et on préfère manipuler des arborescences !

Notons $\mathcal{L}(G,\beta)$ par $\mathcal{L}(\beta)$ et $\mathcal{L}\hat{}(G,\beta)$ par $\mathcal{L}\hat{}(\beta)$

Langages engendrés

Pour tout $\gamma \in (\mathcal{A}+\mathcal{V})^* :\ \gamma \in \mathcal{L}\hat{}(\beta)$ ssi $\beta \overset{*}{\Longrightarrow} \gamma$ ssi $\beta \overset{*}{\longmapsto} \gamma$.
Pour tout $u \in \mathcal{A}^* :\ u \in \mathcal{L}(\beta)$ ssi $\beta \overset{*}{\Longrightarrow} u$ ssi $\beta \overset{*}{\longmapsto} u$.

Le passage aux arborescences permet de démontrer le lemme principal pour les langages algébriques.

Lemme principal

L'application $\mathcal{L}$ qui à chaque $\alpha \in (\mathcal{A}+\mathcal{V})^*$ associe le langage $\mathcal{L}(\alpha) \subseteq \mathcal{A}^*$ est *un morphisme de monoïdes*

$$\mathcal{L} : ((\mathcal{A}+\mathcal{V})^*, \cdot, \varepsilon) \to (\mathcal{P}(\mathcal{A}^*), \cdot, \varepsilon).$$

Par la propriété principale de $(\mathcal{A}+\mathcal{V})^*$, ceci signifie que $\mathcal{L}$ est entièrement déterminée par sa restriction

$$\mathcal{A}+\mathcal{V} \to \mathcal{P}(\mathcal{A}^*)$$

donc, en se rappelant que $\mathcal{L}(x) = x$ pour tout $x \in \mathcal{A}$, par sa restriction

$$\mathcal{L} : \mathcal{V} \to \mathcal{P}(\mathcal{A}^*)$$

La séquentialisation d'une arborescence "arbre par arbre" permet de vérifier cette propriété pour les langages étendus, c'est-à–dire :

- $\mathcal{L}\hat{}(\varepsilon) = \varepsilon$
- $\mathcal{L}\hat{}(\beta\xi) = \mathcal{L}\hat{}(\beta)\mathcal{L}\hat{}(\xi)$ pour tout $\beta \in (\mathcal{A}+\mathcal{V})^*$ et tout $\xi \in \mathcal{A}+\mathcal{V}$.

Il est alors facile d'en déduire les propriétés analogues pour les langages engendrés (non étendus). En fait, on peut préciser un peu ce résultat, en vue de son application pratique.

L'arborescence de $d : \beta\xi \overset{k}{\Longrightarrow} \gamma'$ est un mot sur $\mathcal{D}(G)$ qui se présente sous la forme **ua** : $\beta\xi \overset{k}{\Longmapsto} \gamma\delta$ où $\gamma\delta = \gamma'$ et :

- **u** : $\beta \overset{i}{\Longmapsto} \gamma$ est l'arborescence d'une dérivation $g : \beta \overset{i}{\Longrightarrow} \gamma$,
- **a** : $\xi \overset{j}{\Longmapsto} \delta$ est l'arbre d'une dérivation $h : \xi \overset{j}{\Longrightarrow} \delta$,

avec, bien entendu $i + j = k$.

A partir de cette décomposition, on peut reconstituer, des dérivations équivalentes à d, savoir :

- la composée de $g_1 : \beta\xi \overset{i}{\Longrightarrow} \gamma\xi$ et de $h_1 : \gamma\xi \overset{j}{\Longrightarrow} \gamma\delta$,

ou

- la composée de $h_2 : \beta\xi \overset{j}{\Longrightarrow} \beta\delta$ et de $g_2 : \beta\delta \overset{i}{\Longrightarrow} \gamma\delta$,

dont l'arborescence est évidemment **ua**.

Exemple 1 (suite).

Regardons le cas de la dérivation $d : SbS \overset{10}{\Longrightarrow} aabbabaabba$ dont l'arborescence est à la figure précédente :

$$d_1 : S \overset{7}{\Longrightarrow} aabbabaa$$
$$d_2 : b \overset{0}{\Longrightarrow} b$$
$$d_3 : S \overset{3}{\Longrightarrow} ba$$

sont des séquentialisations respectives des trois arbres de cette figure. On peut construire une dérivation équivalente à d, en composant, par exemple

$$d'_1 : SbS \overset{7}{\Longrightarrow} aabbabaabS$$
$$d'_2 : aabbabaabS \overset{0}{\Longrightarrow} aabbabaabS$$
$$d'_3 : aabbabaabS \overset{3}{\Longrightarrow} aabbabaabba$$

dans cet ordre.

Le lemme principal est rarement cité explicitement. Cependant, on peut indiquer son usage par des expressions comme :

- $d : \beta \overset{k}{\Longrightarrow} \gamma$ se *décompose* en des $d_j : \eta_j \overset{k_j}{\Longrightarrow} \gamma_i, \ldots$
- les $d_j : \eta_j \overset{k_j}{\Longrightarrow} \gamma_i$ se *recomposent* en $\eta_1 \ldots \eta_n \overset{k}{\Longrightarrow} \gamma_1 \ldots \gamma_n$.

Notons que le lemme principal permet de faire des raisonnements par induction lorsque les k_j sont strictement inférieurs à k.

3.1 – Système d'équations associé à une grammaire.

Soit $G = (\mathcal{V}, \mathcal{A}, R)$ une grammaire. Nous avons constaté que G était équivalente à la donnée de la substitution

$$\mathbf{l} : \mathcal{V} \to (\mathcal{A} + \mathcal{V})^*$$

définissant les seconds membres de ses règles globales.

En assimilant les variables à des inconnues (en langages sur $\mathcal{A}$), l'ensemble des équations

$$X = \mathbf{l}(X) \text{ pour toute } X \in \mathcal{V}$$

est appelé **le système d'équations associé à** G.

Une solution (c'est–à–dire, la donnée pour toutes les $X \in \mathcal{V}$, de valeurs $s(X) \subseteq \mathcal{A}^*$ qui vérifient les égalités que le système exprime) est une substitution

$$s : \mathcal{V} \to \mathcal{P}(\mathcal{A}^*)$$

vérifiant l'égalité

$$s(X) = s(\mathbf{l}(X)) \text{ pour chaque } X \in \mathcal{V}$$

(on a étendu s en posant $s(x) = x$ pour tout $x \in \mathcal{A}$).

Résolution du système associé à G

La substitution $\mathcal{L} : \mathcal{V} \to \mathcal{P}(\mathcal{A}^*)$ est la plus petite solution du système associé à la grammaire G.

☞ $\mathcal{L}(X) = \mathcal{L}(\mathbf{l}(X))$ pour toute $X \in \mathcal{V}$.

Les équivalences suivantes sont de simples applications des définitions : pour tout $u \in \mathcal{A}^*$:

$$\begin{aligned} u \in \mathcal{L}(X) \quad & \text{ssi} \quad X \stackrel{*}{\Longrightarrow} u \\ & \text{ssi} \quad \text{il existe } \alpha \in (\mathcal{A}+\mathcal{V})^* \\ & \qquad \text{tel que } (X \longrightarrow \alpha) \in R \text{ et } \alpha \stackrel{*}{\Longrightarrow} u \\ & \text{ssi} \quad \text{il existe } \alpha \in \mathbf{l}(X) \text{ tel que } \alpha \stackrel{*}{\Longrightarrow} u \\ & \text{ssi} \quad u \in \mathcal{L}(\mathbf{l}(X)) \end{aligned}$$

on a donc bien l'égalité annoncée.

☞ Soit $M : \mathcal{V} \to \mathcal{P}(\mathcal{A}^*)$ telle que $M(X) = M(\mathbf{l}(X))$ pour toute $X \in \mathcal{V}$, il faut montrer que $\mathcal{L}(X) \subseteq M(X)$ pour toute $X \in \mathcal{V}$.

L'hypothèse sur M implique que $M(\mathbf{l}(X)) \subseteq M(X)$ pour toute $X \in \mathcal{V}$, c'est–à–dire

$$M(\alpha) \subseteq M(X) \text{ pour tout } \alpha \in \mathbf{l}(X).$$

Montrons, pour tout $u \in \mathcal{L}(X)$, par induction sur la longueur des dérivations $X \stackrel{1}{\Longrightarrow} \alpha \stackrel{k}{\Longrightarrow} u$, que $u \in M(X)$:

- si $k = 0$ alors $\alpha = u$, et donc
$$u = M(u) = M(\alpha) \subseteq M(X),$$
- sinon, le lemme principal permet de décomposer $\alpha \stackrel{k}{\Longrightarrow} u$ en dérivations de longueurs strictement inférieures à $k+1$:
l'hypothèse d'induction permet alors de déduire que $u \in M(\alpha)$, et donc que $u \in M(X)$.

Exemple 1 (suite).

Le système de la grammaire de notre exemple est réduit à l'unique équation $S = SbS + \varepsilon + a + aa$. En appliquant le lemme principal, on peut écrire :

$$\begin{aligned}\mathcal{L}(\mathbf{l}(S)) &= \mathcal{L}(SbS + \varepsilon + a + aa)\\ &= \mathcal{L}(SbS) + \mathcal{L}(\varepsilon) + \mathcal{L}(a) + \mathcal{L}(aa)\\ &= \mathcal{L}(S)b\mathcal{L}(S) + \varepsilon + a + aa\end{aligned}$$

L'égalité $\mathcal{L}(S) = \mathcal{L}(\mathbf{l}(S))$ signifie donc bien que

$$\mathcal{L}(S) = \mathcal{L}(S)b\mathcal{L}(S) + \varepsilon + a + aa.$$

3.2 – Définitions inductives.

La plus petite solution que l'on vient d'obtenir est aussi la plus petite solution du *système d'inéquations* associé à G :

$$\mathbf{l}(X) \subseteq X,$$

car, dans la démonstration précédente, on a seulement utilisé le fait que M vérifiait ce système d'inéquations.

On peut énoncer cette propriété sous la forme d'une définition inductive simultanée des langages $\mathcal{L}(X)$.
Par exemple, dans le cas d'une grammaire ne comportant qu'une seule variable :

$\mathcal{L}(X)$ est le plus petit langage

- qui contient des éléments donnés (ceux de $c(X) = \mathbf{l}(X) \cap \mathcal{A}^*$),
- qui est stable par des opérations données (celles décrites par les éléments de $v(X) = \mathbf{l}(X) - c(X)$).

Exemple 1 (suite).

Soit $L \subseteq (a+b)^*$ le langage constitué des mots qui ne comportent pas le facteur aaa : on sait que ce langage est régulier, nous allons montrer qu'il peut être engendré par une grammaire (non linéaire) traduisant une définition inductive.

- Les deux propriétés suivantes sont immédiates :
  1) $\varepsilon \in L$, $a \in L$ et $aa \in L$, $\qquad (\varepsilon + a + aa \subseteq L)$
  2) si $u_1 \in L$ et $u_2 \in L$ alors $u_1 b u_2 \in L \qquad (LbL \subseteq L)$
- Tous les éléments de L sont obtenus de cette façon : soit $u \in L$
  - si $|\,u\,|_b = 0$ alors $u \in \varepsilon + a + aa$;
  - sinon, en choisissant une occurrence quelconque de b dans u on obtient une décomposition $u = u_1 b u_2$ où $u_1 \in L$ et $u_2 \in L$.

Ces deux points signifient que :

L est le plus petit langage sur l'alphabet $\mathcal{A} = a+b$ vérifiant les propriétés 1) et 2).

On vient de donner une définition inductive de L et ceci est suffisant pour prouver que L est le langage engendré par la grammaire dont l'unique règle globale est

$$S \longrightarrow SbS + \varepsilon + a + aa$$

4 – Transformations des grammaires.

On se souvient qu'un *langage algébrique* $L \subseteq \mathcal{A}^*$ est un langage engendré par une grammaire $G = (\mathcal{V}, \mathcal{A}, R)$ à partir d'une variable (appelée "axiome") $S \in \mathcal{V}$, c'est–à–dire $L = \mathcal{L}(G, S)$.

• G définit un procédé de calcul : l'application des règles permet de produire chaque mot $u \in L$ comme résultat d'une dérivation $S \underset{G}{\overset{k}{\Longrightarrow}} u$.

• Réciproquement, si $u \in \mathcal{A}^*$, la question se pose de savoir si $u \in L$: une réponse affirmative ne sera satisfaisante que si elle est accompagnée de sa preuve, c'est–à–dire, d'une dérivation $S \underset{G}{\overset{k}{\Longrightarrow}} u$. De même, une réponse négative ne sera satisfaisante que si l'on prouve que toute tentative raisonnable de construire une telle dérivation échoue.

Ces questions, synthèse (*synthesis*) et analyse (*parsing*) syntaxiques, justifient que l'on s'intéresse à des grammaires de formes particulières : il s'agit alors de voir s'il est possible de transformer une grammaire donnée en une grammaire de la forme particulière souhaitée, qui soit capable d'engendrer le même langage.

4.1 – Grammaires réduites.

A. Restriction de l'alphabet des variables.

Soit $\mathcal{V}' \subseteq \mathcal{V}$.

La restriction de G *à* $\mathcal{V}'$ est la grammaire $G' = (\mathcal{V}', \mathcal{A}, R')$ où R' est l'ensemble des règles de G qui peuvent s'écrire avec des variables de $\mathcal{V}'$:

$$(X \longrightarrow \alpha) \in R' \text{ ssi } (X \longrightarrow \alpha) \in R$$

pour tout $X \in \mathcal{V}'$ et tout $\alpha \in (\mathcal{A} + \mathcal{V}')^*$.

On a

$$\mathcal{L}(G', X) \subseteq \mathcal{L}(G, X)$$

pour toute $X \in \mathcal{V}'$, car toute règle de G' est aussi une règle de G.

Toutes les parties de $\mathcal{V}$ que nous serons amenés à définir par la suite sont construites de la façon suivante.

- On définit une suite $\mathcal{U}_i$ croissante de parties de $\mathcal{V}$:

$$\mathcal{U}_0 \subseteq \mathcal{U}_1 \subseteq \cdots \subseteq \mathcal{U}_i \subseteq \cdots \subseteq \mathcal{V}.$$

- Une telle suite est stationnaire car $\mathcal{V}$ est fini : il existe un entier N tel que $i \geq N$ implique $\mathcal{U}_i = \mathcal{U}_N$.
- $\mathcal{U} = \mathcal{U}_N$ s'appelle *la limite* de la suite $\mathcal{U}_i$.

B. Variables productives.

- $X \in \mathcal{V}$ est *productive* ssi $\mathcal{L}(G, X) \neq \emptyset$.

$\mathcal{P}rod(G) \subseteq V$ désignera l'ensemble des variables productives de G.

$\mathcal{P}rod(G)$

$\mathcal{P}rod(G)$ est la limite de la suite définie par :

$$\mathcal{U}_0 = \emptyset$$
$$\mathcal{U}_{i+1} = \mathcal{U}_i + \{X \in \mathcal{V} \mid \text{il existe } (X \longrightarrow \alpha) \in R \text{ telle que } \alpha \in (\mathcal{A} + \mathcal{U}_i)^*\}.$$

Elimination des variables non productives

La restriction $G' = (\mathcal{V}', \mathcal{A}, R')$ de G à $\mathcal{V}' = \mathcal{P}rod(G)$ vérifie

- *toutes ses variables sont productives*
- *$\mathcal{L}(G', X) = \mathcal{L}(G, X)$ pour toute $X \in \mathcal{V}'$.*

C. Variables accessibles depuis une variable.

- $X \in \mathcal{V}$ est *accessible depuis* $S \in \mathcal{V}$ ssi
$$\mathcal{L}\hat{}(G, S) \cap [X] \neq \emptyset$$

où $[X] \subseteq (\mathcal{A} + \mathcal{V})^*$ est l'ensemble des α tels que $|\alpha|_X > 0$. $\mathcal{A}cc_G(S) \subseteq \mathcal{V}$ désignera l'ensemble des variables accessibles à partir de S.

On a $S \in \mathcal{A}cc_G(S)$ grâce à la dérivation triviale $S \overset{0}{\underset{G}{\Longrightarrow}} S$.

$\mathcal{A}cc_G(S)$

$\mathcal{A}cc_G(S)$ est la limite de la suite définie par :

$$\mathcal{U}_0 = S$$
$$\mathcal{U}_{i+1} = \mathcal{U}_i + \{X \in \mathcal{V} \mid \text{il existe } Y \in \mathcal{U}_i \text{ et } \alpha \in [X] \text{ tels que } (Y \longrightarrow \alpha) \in R\}.$$

Partie accessible depuis une variable

Soit $S \in \mathcal{V}$ telle que $\mathcal{L}(G, S) \neq \emptyset$.
La restriction $G' = (\mathcal{V}', \mathcal{A}, R')$ de G à $\mathcal{V}' = \mathcal{A}cc_G(S)$ vérifie

- *toute $X \in \mathcal{V}'$ est accessible à partir de S*
- *$\mathcal{L}(G', X) = \mathcal{L}(G, X)$ pour toute $X \in \mathcal{V}'$.*

D. Grammaires réduites.

• G est *réduite pour* $S \in \mathcal{V}$ ssi toutes ses variables sont productives et accessibles à partir de S.

Réduction des grammaires

Pour toute $S \in \mathcal{V}$ *telle que* $\mathcal{L}(G,S) \neq \emptyset$ *il existe une grammaire* $G' = (\mathcal{V}', \mathcal{A}, R')$ *telle que*

- $S \in \mathcal{V}' \subseteq \mathcal{V}$,
- G' *est réduite pour* S,
- $\mathcal{L}(G', X) = \mathcal{L}(G, X)$ *pour toute* $X \in \mathcal{V}'$.

Il suffit d'éliminer les variables non productives puis les variables non accessibles.

Attention : l'élimination des variables qui ne sont pas productives peut rendre d'autres variables inaccessibles à partir de S, il faut donc effectuer les opérations d'élimination dans cet ordre pour réduire G.

4.2 – Grammaires propres.

E. Production de ε.

- *G produit* ε ssi il existe $X \in \mathcal{V}$ tel que $\varepsilon \in \mathcal{L}(G, X)$.
- L'ensemble $\mathcal{E}ps(G) \subseteq \mathcal{V}$ des variables produisant ε est défini par $X \in \mathcal{E}ps(G)$ ssi $\varepsilon \in \mathcal{L}(G, X)$.

$\mathcal{E}ps(G)$

$\mathcal{E}ps(G)$ est la limite de la suite définie par :

$$\mathcal{U}_0 = \emptyset$$
$$\mathcal{U}_{i+1} = \mathcal{U}_i + \{X \in \mathcal{V} \mid \text{il existe } \alpha \in \mathcal{U}_i^* \text{ tel que } (X \longrightarrow \alpha) \in R\}.$$

Elimination des productions de ε

Il existe une grammaire $G' = (\mathcal{V}, \mathcal{A}, R')$ qui ne produit pas ε et qui vérifie $\mathcal{L}(G', X) = \mathcal{L}(G, X) - \varepsilon$ pour toute $X \in \mathcal{V}$.

On ne peut pas se contenter de supprimer les règles de la forme $X \longrightarrow \varepsilon$. Considérons par exemple, la grammaire $G = (S + A, a + b, R)$ dont les règles globales sont :

$$S \longrightarrow SAS + b$$
$$A \longrightarrow a + \varepsilon$$

Grâce à l'existence de la dérivation

$$S \overset{1}{\Longrightarrow} SAS \overset{1}{\Longrightarrow} bAS \overset{1}{\Longrightarrow} bS \overset{1}{\Longrightarrow} bb$$

on a $bb \in \mathcal{L}(G, S)$, mais il est clair que si l'on modifie G en G' en éliminant la règle $A \longrightarrow \varepsilon$, alors $bb \notin \mathcal{L}(G', S)$!

Construction de G'.

- Considérons la substitution $s : \mathcal{A} + \mathcal{V} \to \mathcal{P}((\mathcal{A} + \mathcal{V})^*)$

définie par $s(\xi) = \begin{cases} \xi + \varepsilon & \text{si } \xi \in \mathcal{E}ps(G) \\ \xi & \text{sinon} \end{cases}$ qui remplace chaque variable $X \in \mathcal{E}ps(G)$ par $X + \varepsilon$.

- Si la règle *globale* de X dans G est $X \underset{G}{\longrightarrow} \mathbf{l}$, alors sa règle *globale* dans G' est $X \underset{G'}{\longrightarrow} s(\mathbf{l}) - \varepsilon$.

Exemple 2.

Soit $G = (S + A + B, a + b, R)$ la grammaire définie par les règles globales :

$$S \longrightarrow aAB + BA + b$$
$$A \longrightarrow BBB + a$$
$$B \longrightarrow AB + b + \varepsilon$$

- $\mathcal{E}ps(G) = S + A + B$,
- les règles globales de G' s'écrivent donc :

$$S \longrightarrow (a(A + \varepsilon)(B + \varepsilon) + (B + \varepsilon)(A + \varepsilon) + b) - \varepsilon$$
$$A \longrightarrow ((B + \varepsilon)(B + \varepsilon)(B + \varepsilon) + a) - \varepsilon$$
$$B \longrightarrow ((A + \varepsilon)(B + \varepsilon) + b + \varepsilon) - \varepsilon$$

c'est–à–dire, en développant les seconds membres :

$$S \longrightarrow aAB + aA + aB + a + BA + A + B + b$$
$$A \longrightarrow BBB + BB + B + a$$
$$B \longrightarrow AB + A + B + b$$

F. ε–dérivations.

- Une *ε–dérivation* est une dérivation $X \stackrel{k}{\Longrightarrow} Y$ où $X \in \mathcal{V}$, $Y \in \mathcal{V}$ et $k > 0$.
- La clôture d'une variable X est définie par

$$\mathcal{C}l_G(X) = \{Y \in \mathcal{V} \mid X \stackrel{*}{\underset{G}{\Longrightarrow}} Y\}.$$

On a $X \in \mathcal{C}l_G(X)$ grâce à la dérivation triviale $X \stackrel{0}{\underset{G}{\Longrightarrow}} X$.

Nous supposerons que G ne produit pas ε car alors, une ε–dérivation est composée uniquement d'ε–dérivations élémentaires $Z \stackrel{1}{\underset{G}{\Longrightarrow}} T$.

$\mathcal{C}l_G(X)$

Soit G une grammaire ne produisant pas ε alors, $\mathcal{C}l_G(X)$ est la limite de la suite définie par :

$$\begin{aligned}&\mathcal{U}_0 = X\\ &\mathcal{U}_{i+1} = \mathcal{U}_i + \{Z \in \mathcal{V} \mid \text{il existe } Y \in \mathcal{U}_i\\ &\qquad\qquad\qquad\qquad \text{tel que } (Y \longrightarrow Z) \in R\}.\end{aligned}$$

Par la suite, on notera simplement $\mathcal{C}l(X)$ pour $\mathcal{C}l_G(X)$.

• Pour chaque ensemble de la forme $\mathcal{C}l(X)$ on introduit une variable, désignée par $\overline{X}$:

$$\overline{X} = \overline{Y} \text{ lorsque } \mathcal{C}l(X) = \mathcal{C}l(Y).$$

• Soit $\overline{\mathcal{V}}$ l'ensemble de ces variables, on a une application

$$c : \mathcal{V} \to \overline{\mathcal{V}}$$

définie par $c(X) = \overline{X}$.

Elimination des ε–dérivations

Soit G une grammaire ne produisant pas ε alors, il existe une grammaire $\overline{G} = (\overline{\mathcal{V}}, \mathcal{A}, \overline{R})$ qui n'admet pas d'ε–dérivation, telle que $\mathcal{L}(\overline{G}, \overline{X}) = \mathcal{L}(G, X)$ pour toute $X \in \mathcal{V}$.

La construction des règles gobales de $\overline{G}$ suit la leçon de la détermination des ε–AF.

Soit $X \underset{G}{\longrightarrow} \mathbf{l}(X)$ la règle globale pour $X \in \mathcal{V}$ et considérons l'ensemble

$$\mathbf{m}(X) = \mathbf{l}(X) - \mathcal{V}$$

des éléments de $\mathbf{l}(X)$ qui ne se réduisent pas à une seule variable.

- Si l'on considère les règles globales

$$X \longrightarrow \mathbf{m}(\mathcal{C}l(X))$$

qui contiennent une règle $X \longrightarrow \beta$ pour chaque dérivation

$$X \overset{k}{\underset{G}{\Longrightarrow}} Y \overset{1}{\underset{G}{\Longrightarrow}} \beta$$

où $Y \in \mathcal{V}$ et $\beta \notin \mathcal{V}$, on obtient une grammaire G' qui n'a plus d'ε–dérivation et vérifie $\mathcal{L}(G', X) = \mathcal{L}(G, X)$ de façon évidente.

- On applique c pour prendre en compte le fait que deux variables vérifiant $\mathcal{C}l(X) = \mathcal{C}l(Y)$ ont la même règle globale dans G'.

Ceci conduit à utiliser les éléments de $\overline{\mathcal{V}}$ comme variables, et à écrire la règle globale pour $c(X)$ dans $\overline{G}$ sous la forme :

$$c(X) \longrightarrow c(\mathbf{m}(\mathcal{C}l(X))).$$

Exemple 2 (suite).

Reprenons la grammaire obtenue dans l'exemple précédent :

$$S \longrightarrow aAB + aA + aB + a + BA + A + B + b$$
$$A \longrightarrow BBB + BB + B + a$$
$$B \longrightarrow AB + A + B + b$$

- On a

$$\mathcal{C}l(S) = S + A + B$$
$$\mathcal{C}l(A) = \mathcal{C}l(B) = A + B$$

et

$$\mathbf{m}(S) = aAB + aA + aB + BA + a + b$$
$$\mathbf{m}(A) = BBB + BB + a$$
$$\mathbf{m}(B) = AB + b$$

les règles globales de G' sont donc :

$$S \longrightarrow \mathbf{m}(S) + \mathbf{m}(A) + \mathbf{m}(B)$$
$$A \longrightarrow \mathbf{m}(A) + \mathbf{m}(B)$$
$$B \longrightarrow \mathbf{m}(A) + \mathbf{m}(B)$$

c'est–à–dire :

$$S \longrightarrow aAB+aA+aB+BA+BBB+BB+AB+a+b$$
$$A \longrightarrow BBB + BB + AB + a + b$$
$$B \longrightarrow BBB + BB + AB + a + b$$

- Comme $\overline{B} = \overline{A}$, on peut poser $\overline{\mathcal{V}} = \overline{S} + \overline{A}$, et les règles globales de $\overline{G}$ peuvent s'écrire

$$\overline{S} \longrightarrow a\overline{A}\,\overline{A} + a\overline{A} + \overline{A}\,\overline{A} + \overline{A}\,\overline{A}\,\overline{A} + a + b$$
$$\overline{A} \longrightarrow \overline{A}\,\overline{A}\,\overline{A} + \overline{A}\,\overline{A} + a + b.$$

G. Grammaires propres.

• G est *propre* ssi G ne produit pas ε et n'admet pas d'ε–dérivation.

Nettoyage des grammaires

Il existe une grammaire propre $G' = (\mathcal{V}', \mathcal{A}, R')$ *et une application* $c : \mathcal{V} \to \mathcal{V}'$ *qui vérifient* $\mathcal{L}(G', c(X)) = \mathcal{L}(G, X) - \varepsilon$ *pour toute* $X \in \mathcal{V}$.

Il suffit d'appliquer à G successivement la transformation E puis la transformation F.

Procéder dans cet ordre est indispensable pour deux raisons : d'une part la construction F suppose que la grammaire ne produit pas ε, d'autre part, E peut introduire des ε–dérivations !

5 – Lemme d'itération.

Ce lemme (*pumping lemma*) exprime une propriété nécessaire pour qu'un langage soit algébrique : cette propriété n'est pas suffisante pour cela !

Lemme d'itération

Soit L un langage algébrique, alors il existe un entier $N > 0$ tel que si $u \in L$ et $|u| \geq N$, on peut trouver cinq mots u_1, u_2, u_3, v et w satisfaisant les propriétés suivantes :

1) $u = u_1 v u_2 w u_3$;
2) $vw \neq \varepsilon$;
3) $|vu_2w| \leq N$;
4) $u_1 v^k u_2 w^k u_3 \in L$ *pour tout entier naturel* k.

Rappel de quelques propriétés *bien connues* des arbres finis : Soit **A** un arbre de hauteur h et dont les embranchements sont d'ordre au plus égal à l :

- le nombre de feuilles de **A** est au plus égal à l^h ;
- h est la longueur maximale des branches de **A**.

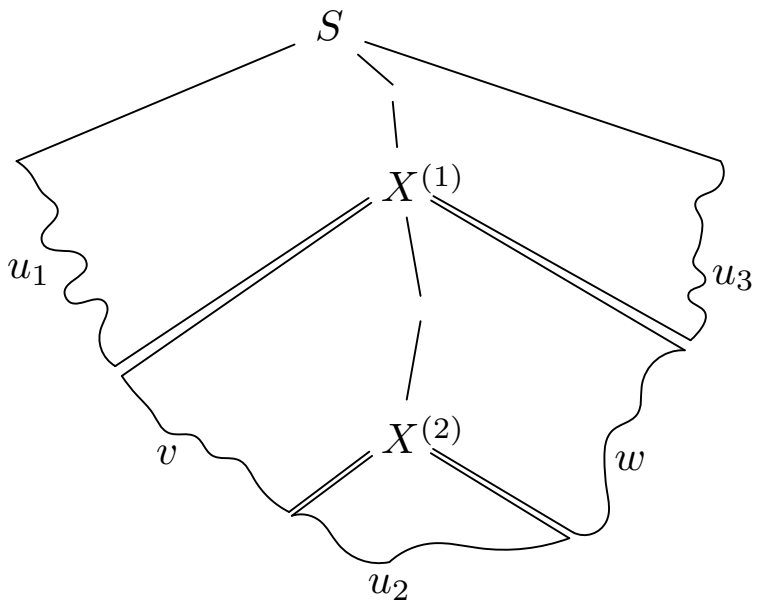
Soient B une branche de **A** de longueur h, X un nœud appartenant à B et **X** le sous–arbre de racine X :

- la partie de B qui se trouve dans **X** est de longueur maximale dans **X** et sa longueur est donc égale à la hauteur de ce dernier.

☞ Si L est fini, il suffit de prendre N strictement supérieur à la longueur maximale des mots de L.

☞ Sinon, soient $G = (\mathcal{V}, \mathcal{A}, R)$ **une grammaire propre**, $S \in \mathcal{V}$ telle que $\mathcal{L}(G, S) = L - \varepsilon$, $V = |\mathcal{V}|$ le nombre de ses variables et l la longueur maximale des α tels que l'on ait $(X \longrightarrow \alpha) \in R$.

Montrons que $N = l^{V+1}$ satisfait le lemme.
Considérons $u \in L$ tel que $|u| \geq N$ et une dérivation $S \Longrightarrow u$: son arbre **A** admet $|u|$ feuilles, sa hauteur est donc au moins égale à $V+1$. Soit B une branche de longueur maximale : elle comporte au moins $V + 1$ nœuds étiquetés par des variables, or celles–ci sont au nombre de V, l'une d'entre elles figure donc au moins deux fois dans B.



Soit X une variable "redoublante" et désignons par $X^{(1)}$ et $X^{(2)}$ deux occurrences distinctes de X figurant dans B, la première choisie plus proche de la racine que la seconde. En élagant **A** d'abord en $X^{(2)}$ puis en $X^{(1)}$ on obtient une dérivation composée

$$S \Longrightarrow u_1 X^{(1)} u_3 \stackrel{n}{\Longrightarrow} u_1 v X^{(2)} w u_3 \Longrightarrow u_1 v u_2 w u_3 = u.$$

• La dérivation intermédiaire $d : X \stackrel{n}{\Longrightarrow} vXw$ est de longueur $n > 0$ et on a donc $vw \neq \varepsilon$ puisque G est propre.

• En substituant à d la dérivation $d^k : X \stackrel{kn}{\Longrightarrow} v^k X w^k$, définie pour chaque entier k comme la composée de k applications de d, qui se définit par récurrence sur k :

$$\begin{aligned} d^0 &: X \stackrel{0}{\Longrightarrow} X \\ d^{k+1} &: X \stackrel{n}{\Longrightarrow} vXw \stackrel{kn}{\Longrightarrow} v^{k+1} X w^{k+1}, \end{aligned}$$

dans la dérivation précédente, on obtient :

$$S \Longrightarrow u_1 X u_3 \stackrel{kn}{\Longrightarrow} u_1 v^k X w^k u_3 \Longrightarrow u_1 v^k u_2 w^k u_3$$

dont l'existence prouve que $u_1 v^k u_2 w^k u_3 \in L$.

Il nous reste à satisfaire la condition $|vu_2w| \leq N$: parmi les variables redoublantes, désignons par X celle pour laquelle $X^{(1)}$ est le plus près possible de la feuille de B. Aucune autre variable redoublante ne se trouve dans la partie B' de B qui commence en $X^{(1)}$. La longueur de B' est donc au plus égale à $V + 1$, le nombre des feuilles du sous–arbre de racine $X^{(1)}$, c'est–à–dire $|vu_2w|$, est donc au plus égal à $l^{V+1} = N$.

Application.

Ce lemme est un outil très efficace pour montrer qu'un langage n'est pas algébrique ; cependant, il ne peut pas servir à montrer qu'un langage est algébrique car ce lemme est une implication dont **la réciproque est fausse**.

Appliquons le lemme pour vérifier que le langage $L = \{a^m b^m c^m \mid m \geq 0\}$ n'est pas algébrique.

Montrons que pour tout $N > 0$ et tout $u \in L$ tel que $|u| \geq N$, une décomposition satisfaisant 1), 2) et 3) ne peut satisfaire 4).

☞ Soit $u = a^N b^N c^N$ et considérons une décomposition $u = u_1 v u_2 w u_3$ satisfaisant 2) et 3). Il découle de 3) que vu_2w ne peut contenir simultanément une occurrence de a et de c :

- si vu_2w ne contient des occurrences que d'une seule lettre, par exemple b, la propriété 4) produit, pour $k = 0$, un mot $a^N b^m c^N$ où $m < N$: un tel mot n'est pas dans L ;
- sinon
  - si v et w ne comportent chacun que des occurrences d'une seule lettre : on aboutit à une réfutation de 4) du type précédent ;
  - sinon, v par exemple est mixte : $v = a^l b^m$ pour $l > 0$ et $m > 0$, alors $v^2 = a^l b^m a^l b^m$ n'est pas un facteur d'un mot de L.

Ce chapitre se termine donc sur une observation pleine de promesses : **tous les langages ne sont pas algébriques**.