Lemme fondamental (v1)

- Soit G = <X, V, S, P> une grammaire algébrique et $f \in (X \cup V)^*$.

Si l'on factorise f en

$f = f_o S_1 f_1 S_2 \ldots f_{k-1} S_k f_k$, $k \geq 1$, avec $\forall$ i, $f_i \in X^*$ et $S_i \in V$,

alors pour tout mot $g \in (X \cup V)^*$,

$f \rightarrow^* g$ si et seulement si il existe des mots $h_1, h_2, \ldots, h_k \in (X \cup V)^*$ tels que

$g = f_o h_1 f_1 h_2 \ldots f_{k-1} h_k f_k$, avec $\forall$ i, $1 \leq i \leq k$, $S_i \rightarrow^* h_i$

Lemme fondamental (v2)

- Soit G = <X,V, S, P> une grammaire algébrique et $f \in (X \cup V)^*$.

  Si l'on factorise f en

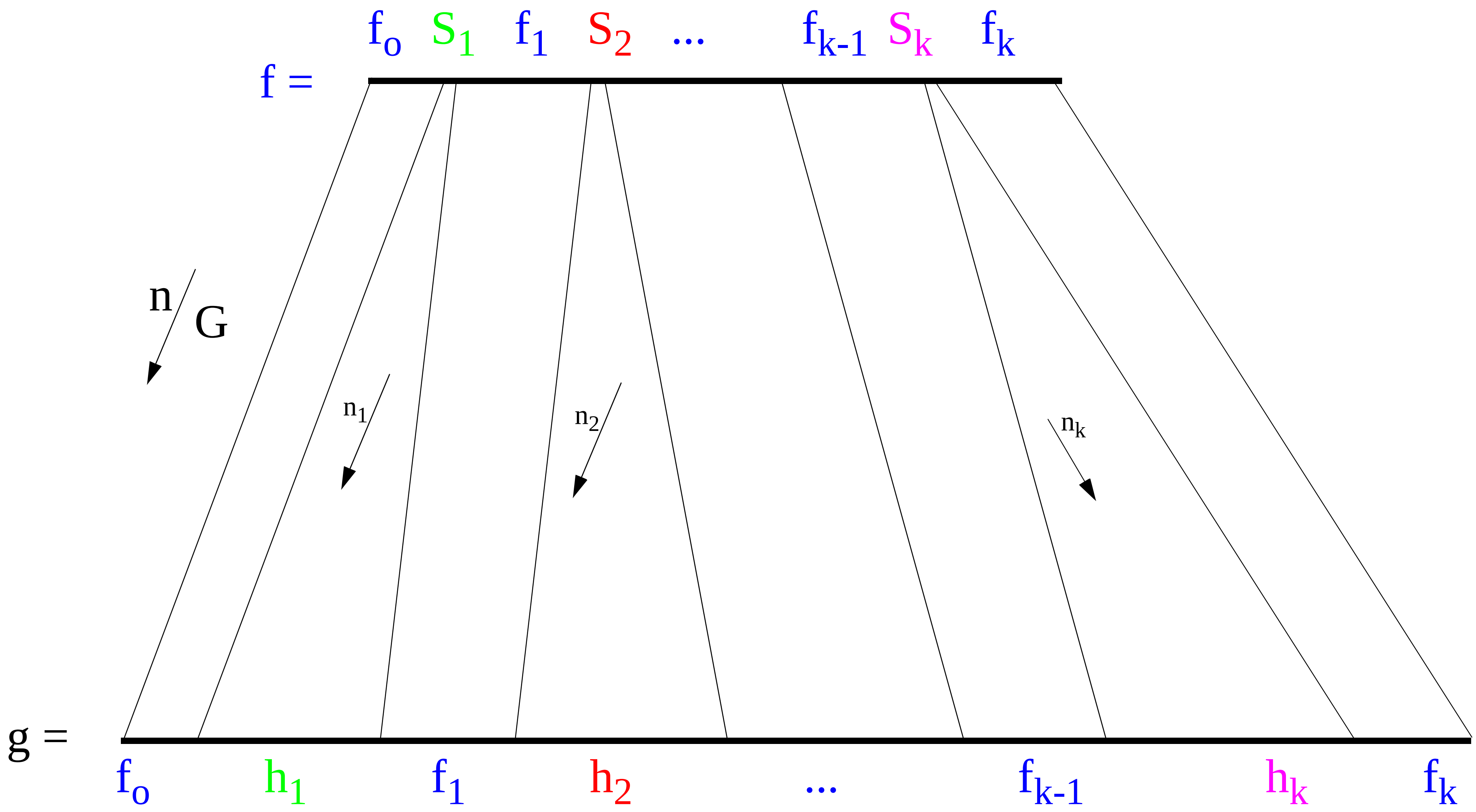
  $f = f_o S_1 f_1 S_2 ... f_{k-1} S_k f_k$, $k \geq 1$, avec $\forall$ i, $f_i \in X^*$, $S_i \in V$, alors

  pour tout entier $n \geq 0$ et pour tout mot $g \in (X \cup V)^*$,

  $f \rightarrow^n g$ si et seulement si il existe des entiers $n_1, n_2, ..., n_k \geq 0$

  et des mots $h_1, h_2, ..., h_k \in (X \cup V)^*$ tels que $n = n_1 + n_2 + ... + n_k$, $g = f_o h_1 f_1 h_2 ... f_{k-1} h_k f_k$ et $\forall$ i, $1 \leq i \leq k$, $S_i \rightarrow^{n_i} h_i$.

- ce lemme se montre par récurrence sur n



$n = n_1 + n_2 + ... + n_k$

Principe de récurrence

- Soit S une partie de IN telle que

1. $0 \in S$,
2. pour tout entier n, si $n \in S$ alors $n+1 \in S$,

alors S = IN.

- **Récurrence simple**

Soit P une propriété définie sur IN, et soit a un entier naturel donné, si

1. P(a) est vraie,
2. $\forall\, n \geq a$, $(P(n) \Rightarrow P(n+1))$ est vraie,

alors P(n) est vraie pour tout entier n supérieur ou égal à a.

Récurrence généralisée

on dit aussi “récurrence forte”, bien qu’elle soit absolument de même puissance que la précédente

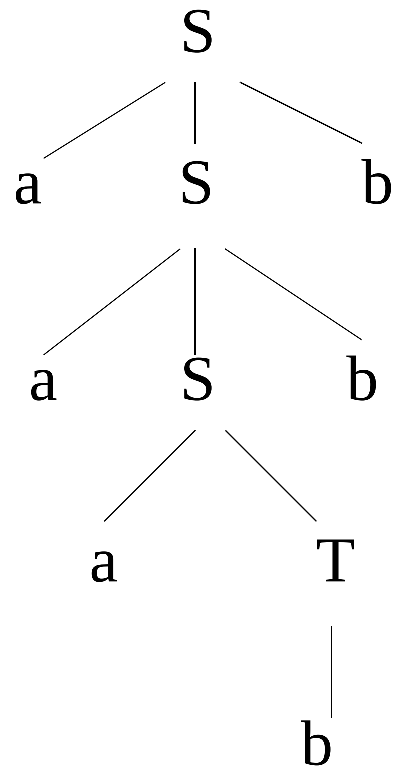
- Soit P une propriété définie sur IN, et soit a un entier naturel donné, si

1. P(a) est vraie,
2. $\forall$ n$\geq$ a, ((P(a) $\wedge$ P(a+1) $\wedge$....$\wedge$ P(n)) $\Rightarrow$ P(n+1)) est vraie,

alors P(n) est vraie pour tout entier n supérieur ou égal à a

Arbre de dérivation

$$G1 : \begin{cases} S \rightarrow aSb \;/\; \; aT \\ T \rightarrow b \end{cases}$$



Arbre de dérivation

Un arbre *A* est un *arbre de dérivation* pour une grammaire algébrique G = < X, V, S, P > si et seulement si :

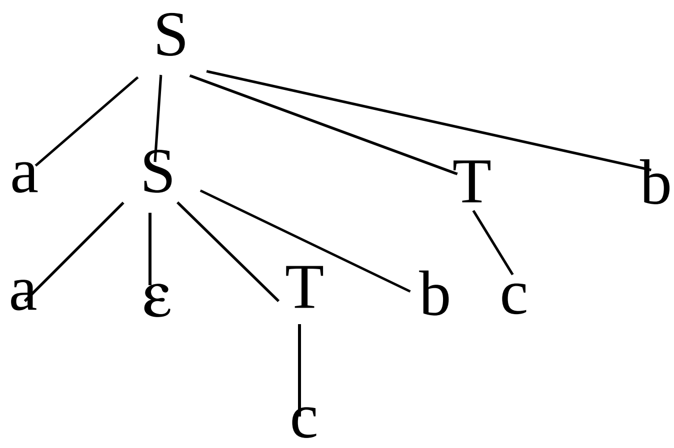
- les étiquettes des nœuds de *A* sont dans $(X \cup V \cup \{\varepsilon\})$
- les nœuds d'étiquette ε ne possèdent pas de frère
- si un nœud de *A* qui n'est pas une feuille est d'étiquette E et si ses descendants directs ont pour étiquettes, de gauche à droite, $e_1, \ldots, e_n$, alors la règle $E \rightarrow e_1 \ldots e_n$ figure dans P

Par conséquent : les nœuds qui ne sont pas des feuilles ont forcément pour étiquette un non-terminal.

Soit G : $S \to a\ S\ T\ b \quad | \quad \varepsilon$

$T \to \quad c$

Quiz 1 - ce qui suit est un arbre de dérivation dans G   vrai   faux



Mot frontière : aacbcb

Quiz 2 - le mot aacbcb est engendré par cette grammaire   vrai   faux

Quiz 3 - tous les mots de $L_G(S)$ possèdent autant de a que de c   vrai   faux

Ambiguïté

- Soit l'instruction if...then...else du langage Pascal, de grammaire

  $S \rightarrow i\ c\ t\ S$ si ... alors

  $S \rightarrow i\ c\ t\ S\ e\ S$ si ... alors ... sinon

  $S \rightarrow a$

  i = if, c = condition, t = then, e = else, a =une instruction

- Soit l'instruction suivante:

  if x=5 then if y>2 then x:=y else x:=z

  i c t i c t a e a

- Ce mot possède deux arbres de dérivation distincts de même racine S.

Dérivation gauche

<phrase> → <groupe-nominal> <verbe> <groupe-nominal>

→ <déterminant> <nom> <verbe> <groupe-nominal>

→ la <nom> <verbe> <groupe-nominal>

→ la souris <verbe> <groupe-nominal>

→ la souris mange <groupe-nominal>

→ la souris mange <déterminant> <nom>

→ la souris mange le <nom>

→ la souris mange le chat

$$G2 : \begin{cases} S \rightarrow U\ V\ b + c \\ V \rightarrow v\ V + v \\ U \rightarrow U\ u + u \end{cases}$$

$$S \rightarrow U\ V\ b \rightarrow U\ u\ V\ b \rightarrow U\ u\ u\ V\ b \rightarrow u\ u\ u\ V\ b \rightarrow u\ u\ u\ v\ V\ b \rightarrow u\ u\ u\ v\ v\ b$$

Un petit point de bilan

on a vu que

G : $\{ S \rightarrow a\,S\,b + \varepsilon$

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

Quiz 1 - le langage $\{a^n b^n, n \geq 10\}$ est engendré par G vrai faux

Quiz 2 - le langage $\{a^n b^n, n \geq 10\}$ est algébrique vrai faux

Quiz 3 - tout langage inclus dans $\{a^n b^n, n \geq 0\}$ est algébrique vrai faux

Étant donnée une supposée grammaire G de L, on doit pouvoir prouver qu'elle engendre effectivement L

- application

$$G : \{ S \rightarrow a\,S\,b + \varepsilon$$

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

- il faut démontrer $L_G(S) = L$ en deux temps :

- $L_G(S) \subseteq L$ (récurrence sur l'ordre de dérivation)

- $L \subseteq L_G(S)$ (récurrence sur la longueur des mots)

(démonstration très détaillée en cours et dans le poly)

Cas particulier 1

- situation avec plusieurs premières règles possibles :

$G_3 : \{ \; S \rightarrow a\, S\, b + S\, b + bb$

engendre le langage $L_3 = \{a^n\, b^p,\ p \geq n+2\}$

$S \rightarrow^{\mathbf{d+1}} f$ se décompose

- soit en $S \rightarrow a\, S\, b \rightarrow^{\mathbf{d}} f$
  on a alors $f = a\, g\, b$ avec $S \rightarrow^{\mathbf{d}} g$ etc ...

- soit en $S \rightarrow S\, b \rightarrow^{\mathbf{d}} f$
  on a alors $f = g\, b$ avec $S \rightarrow^{\mathbf{d}} g$ etc ...

Attention : il y a un seul langage engendré par G

- on pourrait également démontrer que, pour :

$G_3 : \{\ S \rightarrow a\ S\ b + S\ b + bb$ on a $L_{G_3}(S) \subseteq \{a^n\ b^p,\ p \geq n\}$

- mais la réciproque :

$\{a^n\ b^p,\ p \geq n\} \subseteq L_{G_3}(S)$ est fausse !

par exemple, $a^4\ b^5$ n'est pas engendré par cette grammaire

$\{a^n\ b^p,\ p \geq n+2\}$ est l'unique langage engendré par G_3

Cas particulier 2

- situation avec plusieurs non-terminaux :

$$G_2 : \left\{ \begin{array}{l} S \to U\,V\,b + c \\ V \to v\,V + v \\ U \to U\,u + u \end{array} \right.$$

$S \to^{\mathbf{d+1}} f$ se décompose en $S \to U\,V\,b \to^{\mathbf{d}} f$

en vertu du lemme fondamental, on a alors $f = g_1\, g_2\, b$

avec $U \to^{\mathbf{d1}} g_1$ et $V \to^{\mathbf{d2}} g_2$ et $d_1 + d_2 = d$! ! !

Il faut donc examiner les langages engendrés par G_2 à partir de U et de V.

Clôture des algébriques par union

- Si L et M sont algébriques, alors L ∪ M est algébrique.
- Exemple : soit deux grammaires engendrant respect.[t] L et M

$$G \left\{ \begin{array}{l} S \rightarrow a \;\; S \;\; b \;\; + T \\ T \rightarrow T \;\; b + b \end{array} \right.$$

$$G' \left\{ \begin{array}{l} S \rightarrow a \;\; S \;\; b \;\; + T \\ T \rightarrow a\, T + a \end{array} \right.$$

- on construit une grammaire engendrant L ∪ M, ce qui prouve que ce langage est également algébrique