Automates à pile (généraux)

- Un automate à pile est un 6-ou-7-uplet
  $A = < X , Q , Y , q_0, y_0, \lambda, F >$ où :

  - X alphabet d'entrée

  - Q ensemble fini d'états

  - Y alphabet de pile

  - $q_0 \in Q$ est l'état initial

  - $y_0 \in Y$ est le symbole initial de pile

  - λ est une application : $(X \cup \{\varepsilon\}) \times Q \times Y \rightarrow P_{\mathbf{finies}}(Q \times Y^*)$

  - F (facultatif) ensemble d'états finals

Exemple d'automate à pile général

- le premier automate considéré possède 5 règles :

$(X \cup \{\varepsilon\}) \times Q \times Y \quad \rightarrow Q \times Y^*$

(1) a , q_0 , y_o $\rightarrow q_1 , y_0$

(2) a , q_1 , y_o $\rightarrow q_1 , y_0 y_0$

(3) b , q_1 , y_o $\rightarrow q_2 , \varepsilon$

(4) b , q_2 , y_o $\rightarrow q_2 , \varepsilon$

et une epsilon-règle

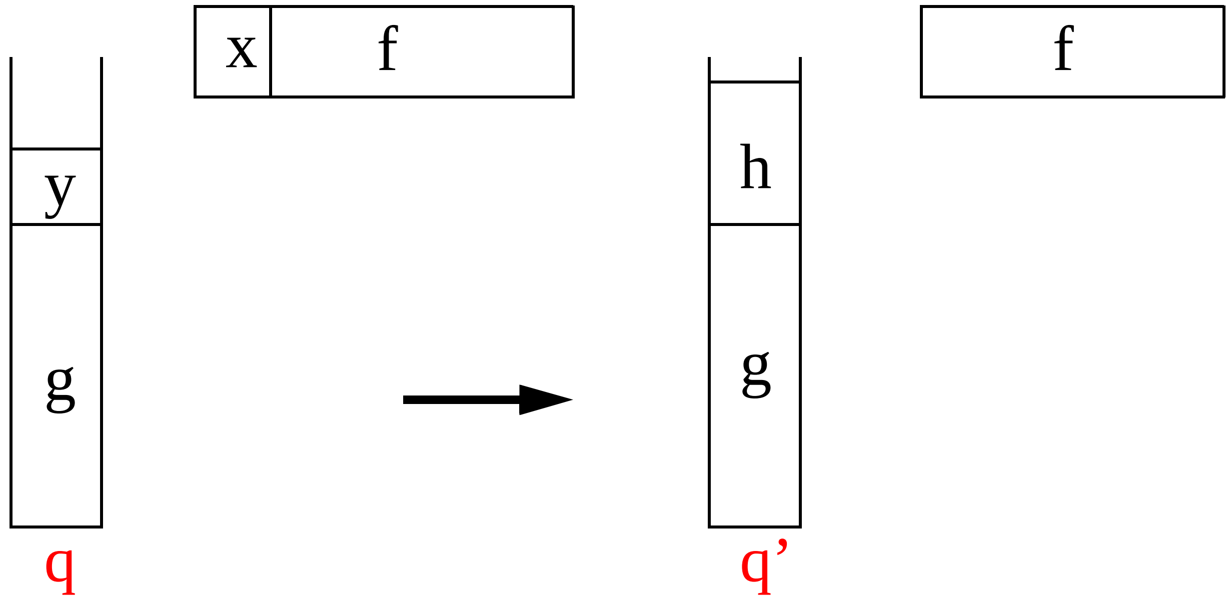
(5) ε , q_0 , y_0 $\rightarrow q_0 , \varepsilon$ $\quad F = \{ q_2 \}$

(6) b , q_2 , y_0 $\rightarrow q_2, y_0$ sera ajoutée ultérieurement

Une configuration = un triplet (f, q, g) de $X^* \times Q \times Y^*$

Transition entre deux configurations

si $x \in (X \cup \{\varepsilon\})$, $q \in Q$, $y \in Y$ et $(q', h) \in \lambda(x, q, y)$

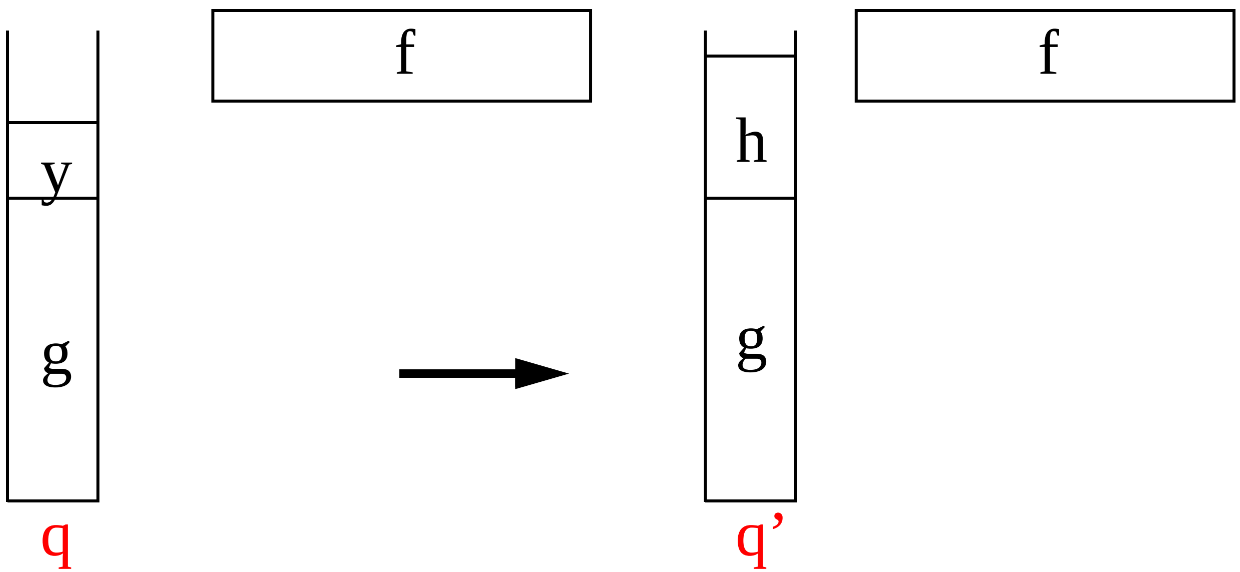


Un calcul valide = une suite de transitions entre configurations

epsilon-transition entre deux configurations

il s’agit juste du cas particulier $x = \varepsilon$

$$\text{si } (q', h) \in \lambda(\varepsilon, q, y)$$



Plusieurs modes de reconnaissance (définition 1)

- par pile vide : N(A) = { $f \in X^*$, il existe un état q quelconque et un calcul valide de (f, q_0, y_0) à $(1\!\!1_X, q, 1\!\!1_Y)$ }

- par états finals : T(A) = { $f \in X^*$, il existe un état q final , un mot g quelconque et un calcul valide de (f, q_0, y_0) à $(1\!\!1_X, q, g)$ }

- d'autres modes existent

donc plusieurs langages reconnus par le même automate, selon le mode choisi.

Deux modes de reconnaissance

(1) $a, \quad q_0, \quad y_o \rightarrow q_1, y_0$

(2) $a, \quad q_1, \quad y_o \rightarrow q_1, y_0y_0$

(3) $b, \quad q_1, \quad y_o \rightarrow q_2, \varepsilon$

(4) $b, \quad q_2, \quad y_o \rightarrow q_2, \varepsilon$

(5) $\varepsilon, \quad q_0, \quad y_0 \rightarrow q_0, \varepsilon$

- par pile vide

Quiz 1 - le langage N(A) est $\{a^n b^n, n \geq 0\}$ $\{a^n b^n, n \geq 1\}$

- par états finals, en prenant $F = \{ q_2 \}$,

Quiz 2 - le mot a^2b^3 est reconnu par états finals vrai faux

Quiz 3 - le mot a^5b^3 est reconnu par états finals vrai faux

Quiz 4 - le langage T(A) est (à proposer par un étudiant) ... vrai faux

Extension de l'application λ

- Soit $\lambda : (X \cup \{\varepsilon\}) \times Q \times Y \to P_{\mathbf{finies}}(Q \times Y^*)$

on l'étend en une application

$\hat{\lambda}$: $X^* \times P(Q \times Y^*) \to P(Q \times Y^*)$ de la façon suivante :

pour $x \in X$, $f \in X^*$, $R \in P(Q \times Y^*)$,

$$\hat{\lambda}(\varepsilon, R) = R \cup \hat{\lambda}(\varepsilon, \{(q', h.w) / (q, y.w) \in R \text{ et } (q', h) \in \lambda(\varepsilon, q, y)\})$$

$$\hat{\lambda}(x.f, R) = \hat{\lambda}(f, \{(q', h.w) / (q, y.w) \in R \text{ et } (q', h) \in \lambda(x, q, y)\})$$

$$\cup \ \hat{\lambda}(x.f, \{(q', h.w) / (q, y.w) \in R \text{ et } (q', h) \in \lambda(\varepsilon, q, y)\})$$

Langages reconnus (définition 2)

- N(A) = $\{ f \in X^*, \exists q \in Q, (q, \varepsilon) \in \hat{\lambda}(f, \{ (q_0, y_0) \}) \}$

- T(A) = $\{ f \in X^*, \exists q \in F, \exists g \in Y^*, (q, g) \in \hat{\lambda}(f, \{ (q_0, y_0) \}) \}$

On passe aux parties infinies

- Soit l'automate

$a\,,\quad q_0\,,\ y_0 \rightarrow q_1\,,\ y_0$

$b\,,\quad q_1\,,\ y_0 \rightarrow q_1\,,\ \varepsilon$

$\varepsilon\,,\quad q_0\,,\ y_0 \rightarrow q_0\,,\ y_0y_0$

$$\hat{\lambda}(\varepsilon\,,\{(q_0\,,y_0)\})$$

$$= \{(q_0\,,y_0)\} \cup \hat{\lambda}(\varepsilon\,,\{(q_0,y_0y_0)\})$$

$$= \{(q_0,y_0),(q_0,y_0y_0)\} \cup \hat{\lambda}(\varepsilon,\{(q_0,y_0y_0y_0)\}) = \ldots.$$

$= \{(q_0,y_0{}^n),\ n > 0\}$, partie **infinie** de $(Q \times Y^*)$

Quiz - le langage N(A) est $\{ab^n,\ n \geq 1\}$ $\{a^nb^p,\ p \geq n \geq 1\}$

Notation plus usuelle (cf automate de la page 2)

- Phase 1 : lecture des a. Dans l'état q_0, on empile un a pour chaque a lu. Après lecture de a^n, la pile contient exactement a^n.
  le changement d'état $q_0 \rightarrow q_1$ devient inutile

- Phase 2 : lecture des b. Au premier b lu, on passe en q_2 . On dépile un a pour chaque b lu.

$$\left.\begin{array}{l} a\,,\quad q_0\,, y_0 \rightarrow q_0\,, a \\ a\,,\quad q_0\,, a \rightarrow q_0, aa \end{array}\right\}\ \text{on empile les a, ce qui les compte}$$

$$\left.\begin{array}{l} b\,,\quad q_0\,, a \rightarrow q_2\,, \varepsilon \\ b\,,\quad q_2\,, a \rightarrow q_2\,, \varepsilon \end{array}\right\}\ \text{on les dépile, ce qui vérifie qu'il y a autant de b que de a}$$

$$\left.\varepsilon\,,\quad q_0\,, y_0 \rightarrow q_0\,, \varepsilon\right\}\ \text{cas } n = 0$$

$N(A) = \{a^n b^n, n \geq 0\}$

Rappel (cas des aps)

- Un langage propre L est algébrique ssi il existe un automate à pile simple qui le reconnaît.

- Un automate à pile simple est un cas particulier d'automate à pile (général) :

  - pas d'état revient à un unique état

  - epsilon-règles interdites

Équivalence grammaires / automates

- Un langage L est algébrique ssi il existe un automate à pile (général) A qui le reconnaît par pile vide, i.e. L = N(A).

sens « seulement si »

- Le cas des algébriques propres est déjà résolu.
- Soit L algébrique non propre, posons $M = L \setminus \{ \varepsilon \}$

  - M est propre (trivial) et algébrique (cf poly p 27)

  - donc M est reconnu par un aps (forcément par pile vide)

  - donc M est reconnu par $A = < X , \{q_0\} , Y , q_0, y_0, \lambda >$
  par pile vide
  sans epsilon-transitions

  - ainsi $L = M \cup \{ \varepsilon \}$ est reconnu par pile vide par

  $B = < X , \{q_0, q_1\} , Y , q_1, y_0,$

  $\lambda \cup \{(\varepsilon, q_1, y_0, q_1, \varepsilon), (\varepsilon, q_1, y_0, q_0, y_0) \} >$

sens « si »

- Soit L = N(A) , il s’agit de construire une grammaire algébrique G qui engendre L, à partir de l’automate dont on dispose.

Ce point est assez délicat : cf « Théorie des langages et automates »,
J.M. Autebert, Masson

Équivalence des 2 modes de reconnaissance

- Si L est le langage reconnu par états finals par un automate à pile A, alors il existe un automate à pile B qui le reconnaît par pile vide.

- Si L est le langage reconnu par pile vide par un automate à pile A, alors il existe un automate à pile B qui le reconnaît par états finals.

- Conséquence : un langage L est algébrique si et seulement si il existe un automate à pile qui le reconnaît par états finals.

- N.B. : pour un langage donné, il y a souvent un mode de reconnaissance plus « naturel » que l'autre.

Non-déterminisme des automates à pile

- Soit à reconnaître par pile vide $L = \{f . \tilde{f} \ , \ f \in \{a, b\}^*\}$

Phase 1 : on lit f et on empile $\tilde{f}$		Phase 2 : on lit $\tilde{f}$ et on dépile $\tilde{f}$	
∗ 1 $a , q_0, y_0 \rightarrow q_0, a$ 2 $b , q_0, y_0 \rightarrow q_0, b$	} lire et empiler première lettre de f	7 $a , q_o, a \rightarrow q_1, \varepsilon$ 8 $b , q_o, b \rightarrow q_1, \varepsilon$	} on commence à dépiler
3 $a , q_0, a \rightarrow q_0, aa$ 4 $a , q_0, b \rightarrow q_0, ab$ 5 $b , q_0, a \rightarrow q_0, ba$ 6 $b , q_0, b \rightarrow q_0, bb$	on empile la suite de f , à l'envers	9 $a , q_1, a \rightarrow q_1, \varepsilon$ 10 $b , q_1, b \rightarrow q_1, \varepsilon$	} on continue de dépiler
		∗ 11 $\varepsilon , q_0, y_0 \rightarrow q_0, \varepsilon$	} cas du mot vide

non déterminisme : 1 ∗ et 11 ∗ ainsi que 6 → 8

Un automate est **déterministe** si tout calcul est entièrement déterminé par l'état, le sommet de pile et la lettre à lire.

Langage déterministe

- On dira d'un langage qu'il est déterministe s'il existe un automate à pile déterministe qui le reconnaît par états finals

  - en particulier, les langages déterministes sont algébriques

- La famille Det(X*) est un sous-ensemble strict de la famille Alg(X*)

- Les langages déterministes sont non-ambigus

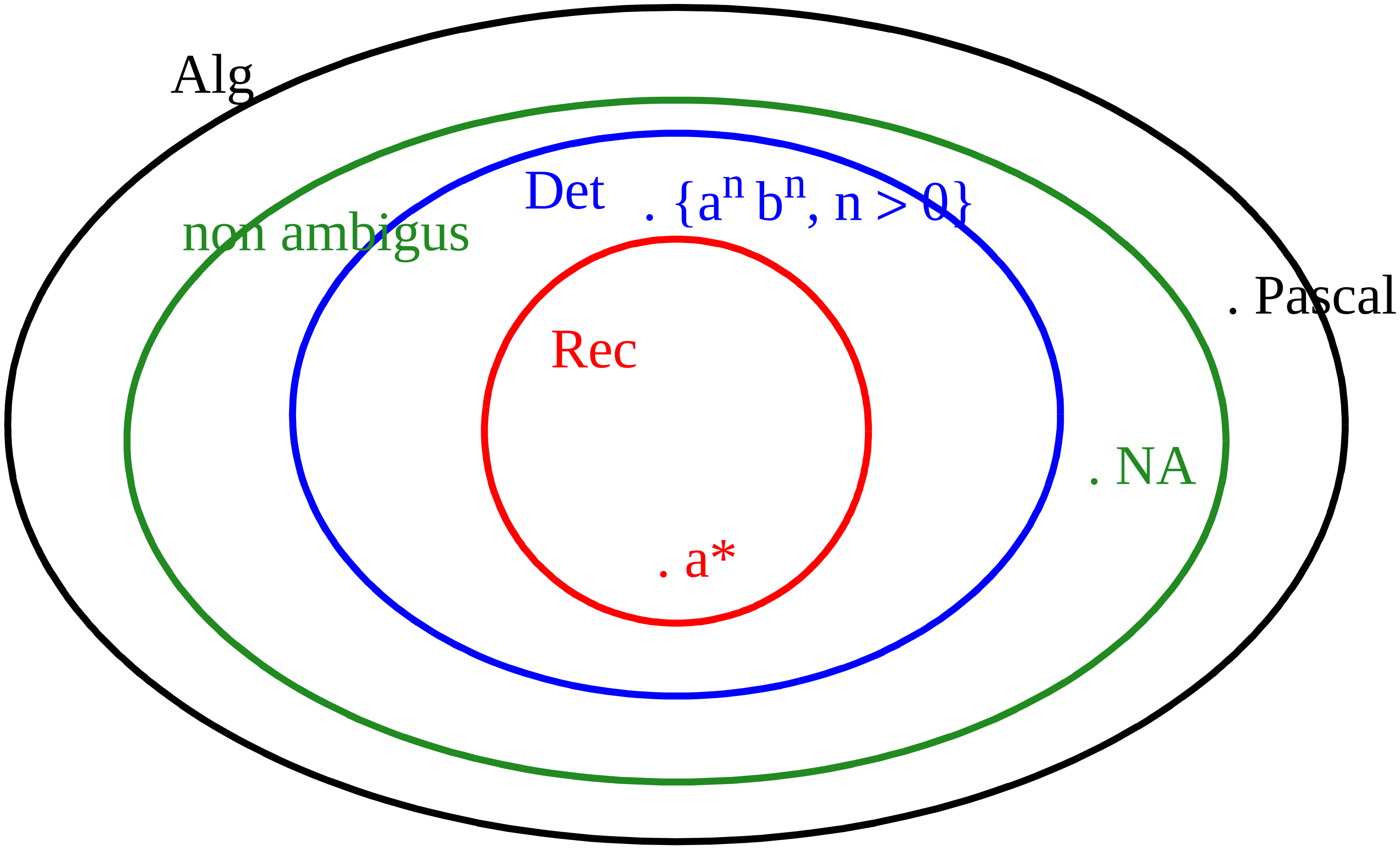
- la réciproque est fausse

  - contre-exemple: $NA=\{a^n b^n, n > 0\} \cup \{a^n b^{2n}, n > 0\}$

non déterministe

non ambigu

Inclusions



Clôture de Alg par intersection rationnelle

- Soit L un langage algébrique et K un langage reconnaissable (i.e. rationnel), $L \cap K$ est alors algébrique.

  - on dit que « Alg est close par intersection rationnelle »

- Rappel : Alg n'est pas close par intersection !

Preuve de cette clôture

- Soit $L = T(A)$, avec $A = < X , Q , Y , q_0, y_0, \lambda, F >$ automate à pile et $K = L(B)$, avec $B = < X' , Q' , q'_0, F' , \delta >$ automate fini,

  alors $L \cap K$ est reconnu par états finals par l'automate à pile :

  $C = < X \cap X', Q \times Q' , Y , (q_0, q'_0), y_0, \gamma, F \times F' >$ avec

  $\gamma = \{(x, (q, r), y, (q', r'), h) \; / \; (x, q, y, q', h) \in \lambda \text{ et } (r, x, r') \in \delta \}$

  $\cup \{(\varepsilon, (q, r), y, (q', r), h) \; / \; (\varepsilon, q, y, q', h) \in \lambda \text{ et } r \in Q' \}$

  donc $L \cap K$ est algébrique !

Corollaire

- La famille Det(X*) est fermée par intersection rationnelle

dans la construction précédente, il suffit de prendre les deux automates A et B déterministes, l’automate résultant C le sera également.

Le théorème de Bar-Hillel, Perles & Shamir

- Si L est un langage algébrique, il existe un entier N tel que tout mot f de L de longueur strictement supérieure à N se décompose en $f = \alpha.u.\beta.v.\gamma$ avec

  - $u.v \neq \varepsilon$,

  - $|u.\beta.v| \leq N$

  - pour tout entier $n \geq 0$, $\alpha.u^n.\beta.v^n.\gamma \in L$.

Lemmes préliminaires

- **Lemme 1** : tout langage algébrique propre possède une grammaire propre
- **Lemme 2** : tout langage algébrique possède une grammaire ne contenant aucune règle $T \rightarrow T'$, et au maximum une règle $T \rightarrow \varepsilon$, avec dans ce cas $T = S$ (l'axiome) (cf poly p 27)
- **Lemme 3** : si f est un « long » mot de $L_G(S)$, tout arbre de dérivation de S en f est « haut », plus précisément, si h est la hauteur d'un tel arbre et p = longueur maximale des membres droits de règles de G, on a $|f| \leq p^h$, d'où $h \geq \log_p(|f|)$
- Par conséquent, si $|f| > p^k$, alors la hauteur de n'importe quel arbre de dérivation de f est > k

Preuve du théorème

- Soit L un langage algébrique, et G = < X, V, S, P > une grammaire de L conforme au lemme 2.

- Soit r = nombre de non-terminaux = |V|,
  p = longueur maximale des membres droits de règles,
  on pose $N = p^{r+1}$. On a $N > 0$ dès que $L \neq \{\varepsilon\}$ car alors $p \neq 0$

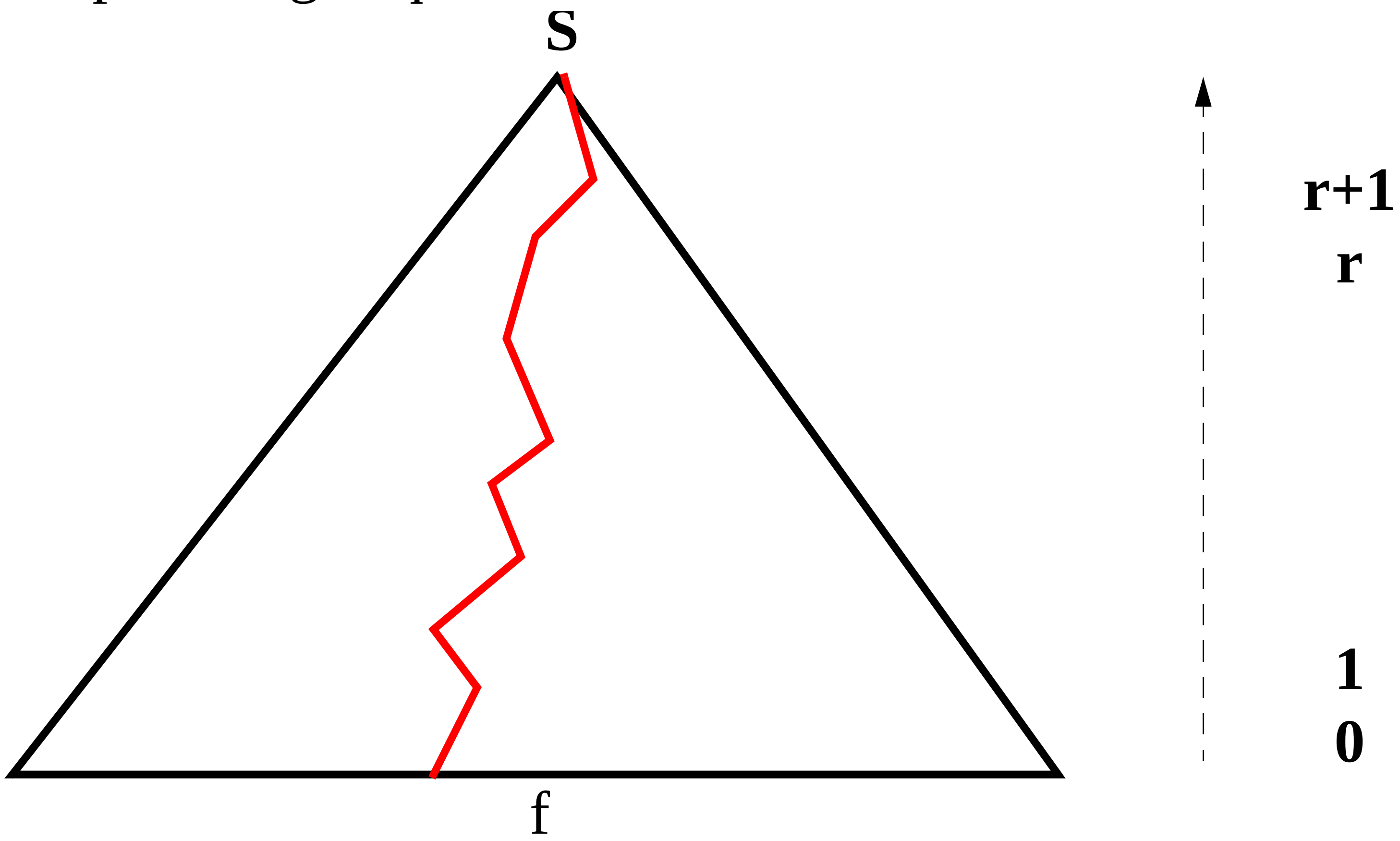
- Soit f un mot de L de longueur $| f | > N = p^{r+1}$

D'après le lemme 3, tout arbre de dérivation de f est de hauteur > r+1.

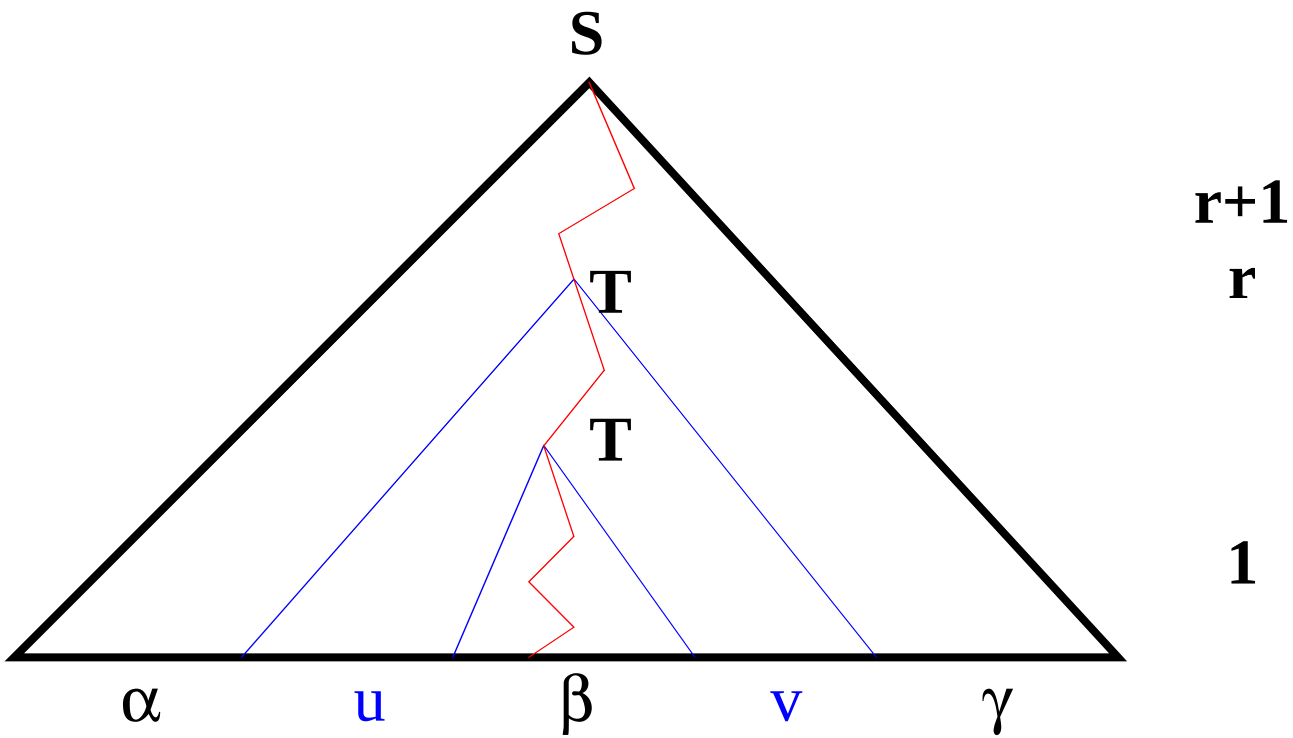
Soit A l'un de ces arbres, sa hauteur est donc au moins égale à r+2

Examinons l'arbre A

- Considérons l'une quelconque parmi les plus longues branches de A, de longueur ≥ r+2. Notons bien que, par ce choix, il n'y a aucune branche dans A plus longue que celle-ci.



- Cette branche possède au moins r+3 étiquettes. La première est S, la dernière est une lettre de X ou ε, les r+1 étiquettes précédant la dernière sont dans V. Ainsi au moins une des ces étiquettes figure 2 fois (notée T sur le schéma).



- On a

$$S \to^* \alpha\, T\, \gamma$$
$$T \to^* u\, T\, v$$
$$T \to^* \beta$$

Vérifions les 3 points du théorème

- Comme la grammaire ne contient aucune règle $U \to U'$ ni $U \to \varepsilon$, on ne peut pas avoir $T \to^{d} T$ avec $d > 0$, et donc $u.v \neq \varepsilon$

- Comme l'arbre A tout entier ne contient aucune branche plus longue que celle choisie, la hauteur du sous-arbre $T \to^{*} u\,\beta\,v$ est au plus r+1. Ainsi, d'après le lemme 3, $|u\,\beta\,v| \leq p^{r+1} = N$

- On peut reproduire aussi souvent que l'on veut la dérivation
  $T \to^{*} u\,T\,v$.
  Ainsi pour tout $n \geq 0$,
  $S \to^{*} \alpha\,T\,\gamma \to^{*} \alpha\,u^{n}\,T\,v^{n}\,\gamma \to^{*} \alpha\,u^{n}\,\beta\,v^{n}\,\gamma$, d'où $\alpha\,u^{n}\,\beta\,v^{n}\,\gamma \in L$.

Application du théorème

- Le langage $\{a^n b^n c^n, n \geq 0\}$ n'est pas algébrique.

  - La preuve est vue en détail en amphi.

- Le langage $L = \{f . f, \; f \in \{a, b\}^* \}$ n'est pas algébrique.