- $f \in L_G(x) \Longleftrightarrow \exists n \in \mathbb{N}$ tq $x \xrightarrow[P]{n} f \in A^*$ rec. sur n

  $n \leq 1 \Rightarrow f = 1$ or $1 \in L_{G'}(x)$ ok/

HR: si $x \xrightarrow[P]{n} f \in A^*$ alors $f \in L_{G'}(x)$

Soit f tq $x \xrightarrow[P]{n+1} f \in A^*$ soit X $\to$ 1 $\xrightarrow{n} f$: $n=0$ $f=1$ ok/

X $\to$ axb $\xrightarrow{n} f = af'b$ $x \xrightarrow{n} f'$

X $\to$ XX

or $f' \in L_{G'}(x)$

$x \to axbx \to af'bx \to af'b = f$

ok/

- Soit $\langle \{a,b\}, \{S,T\}, P \rangle$

  $P: S \to aSbT + b$

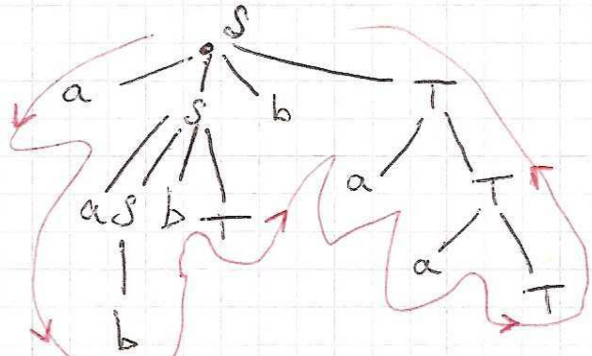
  $T \to aT + a$

Soit la derivation

$S \to aSb\underline{T} \to aSba\underline{T} \to aSbaa\underline{T} \to aa\underline{S}bTaaT$

$\to aabbTaaT$

arbre de derivation :



A un arbre de derivations correspond 1 et 1 seule derivation <u>gauche</u>

La <u>frontière</u> de l'arbre est le mot formé par la suite étiquettes des feuilles de l'arbre vues dans l'ordre lexicographique —

$aabbTbaaT \in \widehat{L}_G(S)$

$f \in \hat{L}_G(s) \Longleftrightarrow \exists$ un arbre de derivation dont la racine est etiqueté par s est dont la frontière vaut f.

Th. G est non ambiguë ssi : $\forall s \in V$, $\forall f \in \hat{L}_G(s)$ $\exists$ un arbre de derivation | racine : s
| frontière $= f$.

Def Une grammaire algébrique $G = \langle A, V, P \rangle$ est dite pré-reduite si $\forall v \in V \quad L_G(v) \neq \emptyset$

Prop Si G gr. alg. alors on peut construire à partir de G, une gram. alg. $G' = \langle A, V', P' \rangle$ avec $V' \subset V$ tele que G' est pré-reduite et $\forall v \in V'$ $L_{G'}(v) = L_G(v)$

Exemple $G = \langle A, V, P \rangle \quad A = \{a, b\}$

P:
$$v_1 \rightarrow v_1 v_2 + v_3 v_4 + v_3 v_5$$
$$v_2 \rightarrow a v_5 + b v_2$$
$$v_3 \rightarrow a v_1 + b v_3 + 1$$
$$v_4 \rightarrow v_2 v_4 + b$$
$$v_5 \rightarrow c v_5 + v_2 v_2$$

1) On construit l'ensemble $\mathcal{U} \subset V$ des variables qui engendrent un langage vide.

Pour cela on calcul le complementaire $\overline{\mathcal{U}}$ de $\mathcal{U}$

$\overline{\mathcal{U}}_0 = \phi$ $\quad \overline{\mathcal{U}}_{i+1} = \overline{\mathcal{U}}_i \cup \{ v \,/\, \exists$ une règle de G $v \to m$ avec $m \in (\overline{\mathcal{U}}_i \cup A)^* \}$

$v\ \overline{\mathcal{U}}_1 = \{v_3, v_4\}$

$\overline{\mathcal{U}}_2 = \{v_3, v_4\} \cup \{v_1\} = \overline{\mathcal{U}}$

$\overline{\mathcal{U}}_3 = \{v_1, v_3, v_4\} \cup \phi$

$\mathcal{U} = \{v_2, v_5\}$

$\overline{\mathcal{U}} = \bigcup_{i \geq 0} \mathcal{U}_i$

Calcul par approximat Π pt fixe.

2) On supprime les variables v_2 et v_5 entierement et toute les règles contenant v_2 ou/et v_5.

donc G' : P':
$$v_1 \to v_3 v_4$$
$$v_3 \to a v_1 + b v_3 + 1$$
$$v_4 \to b$$

~~Def~~ Une gr. alg. $G = \langle A, V, P \rangle$ est dite reduite vis à vis d'un non terminal $v \in V$ si elle est pré-reduite et si toute variable de V peut être atteinte à partir de v dans 1 derivation :

$$\forall v' \in V \ \exists \alpha, \beta \ \text{tq} \ v \xrightarrow{*} \alpha v' \beta$$

Prop Soit $G = \langle A, V, P \rangle$ et $v \in V$ tq $L_G(v) \neq \phi$

On peut construire à partir de G une gram. alg. $G' = \langle A, V', P' \rangle$ reduite vis a vis de v et $L_{G'}(v) = L_G(v)$

même exemple

1) On calcul l'ens $W \subset V$ des variables que l'on peut atteindre à partir de v

$W_0 = \{v\}$

$W_{i+1} = W_i \cup \{v' \in V \mid \exists$ une règle $v \to w$, $v \in W_i$ et w contient v' $\}$

On calcul pt fixe

2) On supprime toute les règles qui contiennent une occurence dans $\overline{W}$ complémentaire de W.

Prop Si $G = \langle A, V, P \rangle$ alors $\forall v \in V$ on peut décider si $1 \in L_G(v)$ ou non

1) On construit l'ens. $S \subset V$ des variables qui peuvent engendrer le mot vide

$S_0 = \{v \in V \mid \exists (v, 1) \in P\}$

$S_{i+1} = S_i \cup \{v \in V \mid \exists (v, m)$, $m \in S_i^*$

On calcul le pt fixe. S

2) $\forall (v, m) \in P$ on substitue dans m a chaque $v \in S$ l'ens $\{v, 1\}$

3) On suppr. toute regle de la forme $(v, 1)$

Prop Si $G = \langle A, V, P \rangle$ alors on peut construire à partir de G, $G' = \langle A, V, P' \rangle$ telle que $\forall v \in V$

$$L_{G'}(v) = L_G(v) \setminus \{1\}$$

Def Un langage $L \subset A^*$ est propre
ssi $1 \notin L$

Exemple $A = \{a, b, c, d\}$

P: $v_1 \to v_2 v_3 + a v_4 b v_5$

$v_2 \to c v_1 + a v_2 v_4 + v_3 v_5$

$v_3 \to v_4 c v_5 + d v_5 + v_5 v_5$

$v_4 \to a v_1 v_2 + v_1 v_1$

1) Calcul

$v_5 \to v_2 a v_3 + 1$

$S_0 = \{v_5\}$ $S_1 = \{v_5, v_3\}$ $S_2 = \{v_5, v_3, v_2\}$

$S_3 = \{v_5, v_3, v_2, v_1\}$ $S_4 = V = S$

2) Substitution

$v_1 \to \{1, v_2\}\{v_3, 1\} + a\{v_4, 1\} b \{v_5, 1\}$

$= v_3 + 1 + v_2 v_3 + v_2 + a v_4 b v_5$

$+ a v_4 b + a b v_5 + ab$ (on dvpt).

$v_2 \to \ldots$

3) On supprime les 1 dans les règles.

Def $G = \langle A, V, P \rangle$ est dite propre ssi elle n'a pas de règles de forme $v \to 1$ ni $v \to v'$

Remq G propre $\Longrightarrow \forall v \in V \; L_G(v)$ propre.

Prop

Si $G = \langle A, V, P \rangle$ alors on peut construire une grammaire Alg. propre $G' = \langle A, V', P' \rangle$ a partir de G tq :

$$\forall v \in V \ \exists v' \in V' \text{ tq } L_{G'}(v') = L_G(v) \setminus \{1\}$$

1) Supprim toute regle $(v, 1)$ par la méthode précédente.

2) On definit R : $(v, v') \in R \Longleftrightarrow \exists (v, v')$

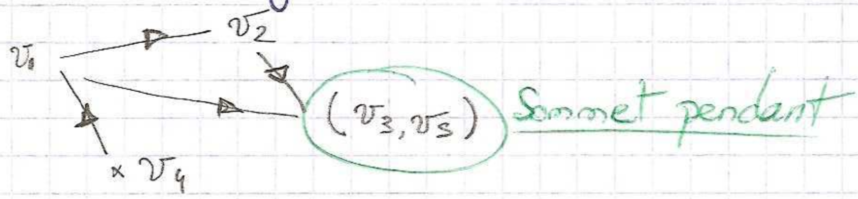
a) On regarde toutes les classes d'equivalence :

(v, v') && (v', v)

On confonds les variables réunies dans une classe en une variable, pour chaque classe.

Cela revient a supprimer les circuits.

$\exists$ dans le nouveau graphe des non terminaux n'ayant pas de fils :



3) On substitue repetitivement aux occurrences seules des variables qui sont <u>sommet pendant</u>, les membres droits issus de ces variables.

Prop

Si $G = \langle A, V, T \rangle$ et $f \in A^*$ alors $\forall v \in V$ on peut decider si $f \in L_G(v)$ ou non

1) si $f = 1$ on applique procedure précedente adequate.

2) $f \in L_G(v)$ ssi $f \in L_G(v) \setminus \{1\}$

On construit G' propre a partir de G

$f \in L_{G'}(v)$?

$\Longleftrightarrow \exists$ dérivation $v \xrightarrow[P']{}^{*} f \Longleftrightarrow \exists n \in \mathbb{N} \quad v \xrightarrow[P']{}^{n} f$

Dans cette dérivation il y a au plus $|f|$ règles terminales qui ont été appliquées

______ au plus $|f|-1$ règles non-terminales qui ont été appliquées

Au plus il y a $2.|f|-1$ règles qui ont été app.

par csq : $\exists n \in \mathbb{N}$ avec $0 < n < 2|f|$ tq
$v \xrightarrow[P']{}^{n} f$.

3) On fait toute les dérivations partant de v de longueur $< 2.|2f|$
On regarde si f figure parmis les mots ainsi engendrés

Def
- $G = \langle A, V, P \rangle$ est dite presque Greinbach si ses règles sont de la forme :
$v \to xm$ avec $x \in A$ et $m \in (A \cup V)^*$

- Normale de Greinbach si ses règles sont de la forme $v \to xm$ avec $x \in A$ et $m \in V^*$

Rmq - Une gr. alg. (presque) de Greinbach est propre.
- Si G presque Greinbach on peut construire G' sous forme de Greinbach.

Exemple $v_1 \to a v_1 \; b v_2 + b$

$v_2 \to ba + a v_1$

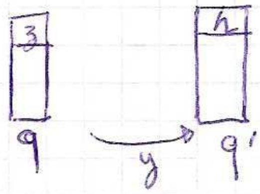
on ajoute $v_a \to a$

$v_b \to b$

Def Un automate à pile $\mathcal{A} = \langle Q, Z, \lambda, i, K \rangle$ est deterministe si :

1) $\forall y \in A \cup \{1\}$, $\forall q \in Q$, $\forall z \in Z$:

$$\| \{ (y, q, z, h, q') \in \lambda \} \| \leqslant 1$$



2) Si $\exists$ une règle $(1, q, z, h, q') \in \lambda$ alors
$\forall x \in A \ \nexists \ (x, q, z, h', q'') \in \lambda$

Rmq ① Si $\mathcal{A}$ deterministe et reconnait L par pile vide (ou pile vide et états d'acceptations) alors

$$\left(f \in L \text{ et } f.g \in L \Longrightarrow g = 1 \right) \Longleftrightarrow L \text{ prefixe}$$

Def L langage prefixe si il verifie :

$$f \in L \text{ et } fg \in L \Longrightarrow fg = 1$$

Def ① Un langage $L \subset A^*$ est <u>deterministe</u> si $\exists$ un automate à pile $\mathcal{A} = \langle Q, Z, \lambda, i, K \rangle$ deterministe qui le reconnait par <u>états d'acceptations</u>

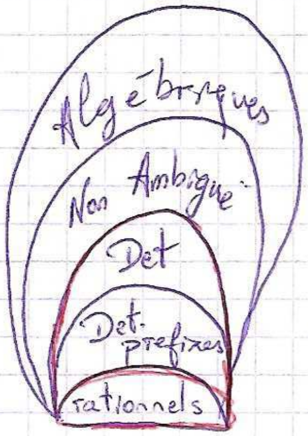
② L est <u>deterministe prefixe</u> si il est reconnu par pile vide par un automate a pile deterministe.

Prop Un langage est deterministe prefixe ssi c'est un langage deterministe qui est prefixe.

Rmq ① Quand on rend un auto. à pile deterministe à fond de pile testable on obtient un automate à pile deterministe.

② Si on fait la construction des triples à partir d'un $\mathcal{A}$ a pile deterministe on obtient une grammaire non ambigue.

Prop Si L est deterministe alors $\overline{L}$ aussi



– fermés par complément