Forme normale quadratique faible

<u>ex:</u> $S \to aSbS \mid \varepsilon$ n'est pas en FNQF
le membre droit $aSbS$ est de longueur >2

$S \to \varepsilon \mid aS_1$
$S_1 \to SS_2$
$S_2 \to bS$

est une grammaire équivalente en FNQF

<u>Lemme</u> Soit $G = (\Sigma, V, P, S)$
On peut construire en temps linéaire G' équivalente en FNQF

En particulier $|G'| = O(|G|)$

$V' \leftarrow V$, $P' \leftarrow \emptyset$

$\forall$ règle $r = (x, \alpha) \in P$ faire $\qquad O(\|P\|)$

- si $|\alpha| \leq 2$ alors
  - ajouter $x \to \alpha$ à P'
- sinon $\alpha = \alpha_1 \alpha_2 \cdots \alpha_k$ $\quad k \geq 3$ $\quad \alpha_i \in \Sigma \cup V$
  - ajouter $r_1, \ldots, r_{k-2}$ à V'
    où $r_1, \ldots, r_{k-2}$ sont des nouvelles variables
  - ajouter à P'
    $x \to \alpha_1 r_1$, $r_1 \to \alpha_2 r_2$, $\ldots$ $r_{k-2} \to \alpha_{k-1} \alpha_k$

Calcul des Variables productives en temps linéaire

$X_0 = \emptyset \qquad X_{n+1} = \{ x \in V \mid \exists\, x \to \alpha \in P \text{ avec } \alpha \in (\Sigma \cup X_n)^* \}$

$X = \bigcup_{n \geq 0} X_n$

__Rem:__ - $(X_n)_{n \geq 0}$ suite $\nearrow$

- $X_n = X_{n+1} \Rightarrow X = X_n$
- $X = X_{|V|}$

__Lemme__ $\forall n \geq 0 \quad X_n = Y_n$ où

$Y_n = \{ x \in V \mid \exists\, t \text{ arbre de dérivation avec } x = rac(t),\ Fr(t) \in \Sigma^*,\ h(t) \leq n \}$

__Preuve:__ Récurrence sur n.

__- $n = 0$__ si $x = rac(t)$ et $h(t) = 0$ alors $Fr(t) = x \notin \Sigma^*$. Donc $Y_0 = \emptyset$ ✓

__- $n \to n+1$__ : $\boxed{\subseteq}$ Soit $x \in X_{n+1}$

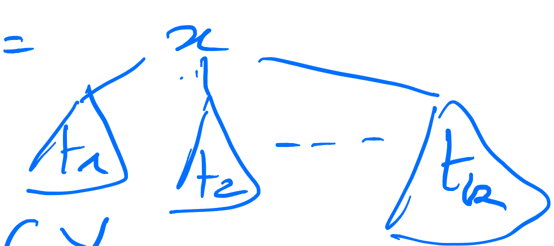
$\exists\, x \to \alpha \in P$ avec $\alpha \in (\Sigma \cup X_n)^*$

$\alpha = \alpha_1 \alpha_2 \cdots \alpha_k$

si $\alpha_i \in \Sigma$ soit $t_i = \alpha_i$

si $\alpha_i \in X_n$ soit t_i un AD de racine α_i, $Fr(t_i) \in \Sigma^*$ et $h(t_i) \leq n$ (HR)

Soit $t =$

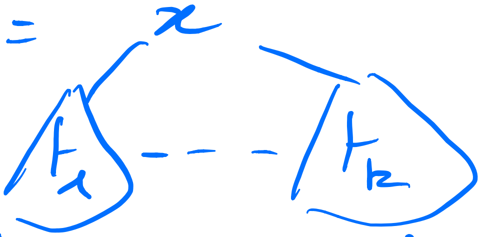


t est 1 AD
$Fr(t) \in \Sigma^*$
$h(t) \leq n+1$

Donc $x \in Y_{n+1}$

$\boxed{\supseteq}$ Soit $x \in Y_{n+1}$ et t un AD tq ...
$Fr(t) \in \Sigma^*$ donc $h(t) \geq 1$

Alors $t =$



avec $\forall i$, t_i AD
$Fr(t_i) \in \Sigma^*$
$h(t_i) \leq n$

- soit $h(t_i) = 0$ et $\alpha_i := rac(t_i) \in \Sigma$ $(\cup\{\varepsilon\})$
- Soit $\alpha_i := rac(t_i) \in Y_n \subseteq X_n$ par HR

Donc $x \to \alpha = \alpha_1 \dots \alpha_k \in P$ et $\alpha \in (\Sigma \cup X_n)^*$ $\square$

<u>Cor</u> : X est l'ensemble des variables productives

<u>Calcul naïf de X</u> : $O(|V| . |G|)$

$\to$ $|V|$ itérations car $X = X_{|V|}$
$\to$ Chaque itération (calcul de X_{n+1}) se fait en $O(\|P\|)$

Réduction à l'évaluation de clauses de Horn.

Pour chaque règle $x \to \alpha \in P$
créer la clause $y_1, \dots, y_k \to x$
où $V \cap alph(\alpha) = \{y_1, \dots, y_k\}$ | $O(|G|)$

En particulier, une règle $x \to \alpha$ avec $\alpha \in \Sigma^*$
engendre la clause $\to x$

Une variable est productive ssi elle s'évalue à Vrai

$\Rightarrow$ On peut calculer les variables productive en temps linéaire.

Calcul de X en $\mathcal{O}(|G|)$: Algo direct

Pour chaque variable $y \in V$ on crée la liste Règles(y) des règles $x \to \alpha \in P$ tq $y \in alph(\alpha)$.

```
CréerFile(V) F;
∀ x ∈ V faire | Prod[x] ← False                      O(|V|)
              | Règles[x] ← ∅
Pour chaque r = (x,α) ∈ P faire                      O(||P||)
  | n[r] ← |V ∩ alph(α)|
  | si n[r] = 0 ∧ ¬Prod[x] alors
  |     Prod[x] ← True ; F.enfiler(x)                O(1+|α|)
  | Sinon
  |     ∀ y ∈ V ∩ alph(α) faire
  |         ajouter r à Règles[y]

TQ F ≠ ∅ faire                                       O(||P||)
  | y ← F.defiler()
  | ∀ r = (x,α) ∈ Règles[y] faire
  |   | n[r] ← n[r] - 1
  |   | si n[r] = 0 ∧ ¬Prod[x] alors
  |   |     Prod[x] ← True ; F.enfiler(x)
```

$\Rightarrow$ On peut décider en temps $\mathcal{O}(|G|)$ si $\mathcal{L}_G(S) = \emptyset$.

Calcul des variables accessibles en temps linéaire

Soit $G = (\Sigma, V, P, S)$

On construit un graphe

Sommets : V

arêtes : $E = \{ (x, y) \mid \exists (x, \alpha) \in P$ avec $y \in \text{alphabet}(\alpha) \}$

Remarques :

1) On peut construire le graphe (V, E) en temps $O(|G|)$

2) Une variable est accessible dans G ssi elle est accessible à partir de S dans le graphe (V, E).

3) On peut calculer l'ensemble Y des variables accessibles par un parcours (largeur ou profondeur) du graphe (V, E) à partir de S.

$\longrightarrow$ temps linéaire.

Réduction d'une grammaire (Linéaire)

- Calculer l'ensemble X des var productives $O(|G|)$

  Calculer $P' = P \cap X \times (\Sigma \cup X)^*$ $O(|G|)$

  $G = (\Sigma, V, P, S)$ équivalente à $G' = (\Sigma, X, P', S)$

- Calculer l'ensemble Y des var accessibles de la grammaire G'. $O(|G|)$

  Calculer $P'' = P' \cap Y \times (\Sigma \cup Y)^*$ $O(|G|)$

  $G'' = (\Sigma, Y, P'', S)$ est réduite et équivalente à G.

Si on restreint aux variables accessibles <u>puis</u> aux variables productives la grammaire obtenue n'est pas forcément réduite.

<u>Ex:</u> $S \to a + S_1 \quad S_1 \to S_1 S_2 \quad S_2 \to b$

Toutes les variables sont accessibles

Mais S_1 n'est pas productive

Si on l'enlève il reste $S \to a \quad S_2 \to b$

et S_2 n'est plus accessible.

Suppression des ε-règles : (Linéaire)

1) Calculer $Z = \{x \in V \mid \exists\, x \xrightarrow{*} \varepsilon\}$

<u>Rem</u> : Z est l'ensemble des variables productives avec les règles $P \cap V \times V^*$

Donc on peut calculer Z en $O(|G|)$

$\rightarrow$ On peut décider en $O(|G|)$ si $\varepsilon \in \mathcal{L}_G(S)$

2) On suppose G en FNQF

$P' \leftarrow \emptyset$

$\forall$ règle $x \to \alpha \in P$ $\quad O(\|P\|)$

- Si $\alpha \neq \varepsilon$ alors ajouter $x \to \alpha$ à P'
- Si $\alpha = \alpha_1 \alpha_2 \wedge \alpha_1 \in Z$ alors ajouter $x \to \alpha_2$ à P'
- Si $\alpha = \alpha_1 \alpha_2 \wedge \alpha_2 \in Z$ alors ajouter $x \to \alpha_1$ à P'

$G' = (\Sigma, V, P')$

<u>Rem</u> Si $S \in Z$ on peut ajouter un nouvel axiome S' et les règles $S' \to S + \varepsilon$

<u>Rem</u> : . G' est encore quadratique

. Si $\mathcal{L}_G(x) = \{\varepsilon\}$ alors $\mathcal{L}_{G'}(x) = \emptyset$ (cf. Rem ci-dessous)

Donc G' n'est pas forcément réduite, même si G est réduite

. La réduction d'une grammaire sans ε-règle est encore sans ε-règle.

Lemme $\forall x \in V \quad \mathcal{L}_{G'}(x) = \mathcal{L}_G(x) \setminus \{\varepsilon\}$

Preuve: $\boxed{\subseteq}$ Facile.

Si $x \to \alpha' \in P'$ alors $x \xrightarrow{+} \alpha'$ dans G.

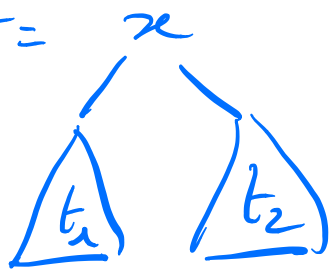
Donc $x \xrightarrow{*} w$ dans $G' \Rightarrow x \xrightarrow{*} w$ dans G.

De plus $\varepsilon \notin \mathcal{L}_{G'}(x)$ car il n'y a pas d'ε-règle dans G'.

$\boxed{\supseteq}$ Rec. sur hauteur arbre dérivation

Soit t AD avec $rac(t) = x$ et $w = Fr(t) \in \Sigma^+$

1er Cas $t =$



Soit $x_i = rac(t_i)$ et $w_i = Fr(t_i)$

Si $w_i \neq \varepsilon$ alors

(HR) $x_i \xrightarrow{*} w_i$ dans G'

- Si $w_1 \neq \varepsilon \neq w_2$ alors $x \to x_1 x_2 \in P'$ et $x \xrightarrow{*} w_1 w_2 = w$ dans G'
- Si $w_1 \neq \varepsilon = w_2$ alors $x \to x_1 \in P'$ et $x \xrightarrow{*} w_1 = w$ dans G'
- Si $w_1 = \varepsilon \neq w_2$ idem
- $w_1 = \varepsilon = w_2$ est impossible car $w_1 w_2 = w \neq \varepsilon$

2ème Cas $t =$

$w = Fr(t_1) \in \Sigma^+$

HR $rac(t_1) \xrightarrow{*} w$ dans G'

De plus $x \to rac(t_1) \in P'$ $\square$

Suppression des règles $x \to y$ $\quad O(|V| \times |G|)$

Soit $G_1 = (\Sigma, V, P_1)$ une grammaire.

Soit $P_2 = \{ (x, \alpha) \mid \exists\, y \to \alpha \in P_1$ avec $\alpha \notin V$ et $x \xrightarrow{*} y$ dans $G_1 \}$

$G_2 = (\Sigma, V, P_2)$ n'a pas de règle $x \to y$

<u>Lemme</u> $\forall x \in V \quad \mathcal{L}_{G_1}(x) = \mathcal{L}_{G_2}(x)$

<u>Preuve</u> $\boxed{\subseteq}$ récurrence longueur dérivation.

- si $x \to w \in \Sigma^*$ dans G_1 alors $x \to w \in P_2$ ✓
- si $x \to x_1 \to \dots \to x_k \to \alpha \xrightarrow{*} w$ dans G_1 avec $\alpha \notin V$ alors $x \xrightarrow{*} x_k$ dans G_1 donc $x \to \alpha \in P_2$. HR $\Rightarrow$ $\alpha \xrightarrow{*} w$ dans G_2 ✓

$\boxed{\supseteq}$ récurrence longueur dérivation

$x \to \alpha \xrightarrow{*} w \in \Sigma^*$ dans G_2.

$\exists\ y \to \alpha \in P_1$ avec $x \xrightarrow{*} y$ dans G_1

De plus (HR) $\alpha \xrightarrow{*} w$ dans G_1

Donc $x \xrightarrow{*} w$ dans G_1 □

<u>Rem</u> : $\|P_2\| = |V| \times \|P_1\|$

- G_1 quadratique $\Rightarrow$ G_2 aussi
- G_1 sans ε-règle $\Rightarrow$ G_2 aussi
- G_2 peut avoir des variables non accessibles
- La réduction d'une grammaire propre (pas de $x \to \varepsilon$ ou $x \to y$) est encore propre

Calcul de P_2 en temps $O(|V| \times |G_1|)$

On suppose que G_1 n'a pas d'ε-règle !

1) $\forall y \in V$ calculer
$$W(y) = \{x \in V \mid x \xrightarrow{*} y \text{ dans } G_1\}$$

<u>Algo</u>
- Construire le graphe (V, E) tq
  $E = \{(y, x) \in V^2 \mid x \to y \in P_1\}$ $\quad O(\|P_1\|)$
- $\forall y \in V$ faire un DFS dans (V, E) à partir de y pour obtenir $W(y)$

DFS(y) : $O(|E|) = O(\|P_1\|)$
$\forall y \in V$: $|V|$ fois $\to O(|V| \times \|P_1\|)$

2) $\forall y \to \alpha \in P_1$ avec $\alpha \notin V$ $\quad O(|V| \times \|P_1\|)$
   $\forall x \in W(y)$
      Ajouter $x \to \alpha$ à P_2

<u>Corollaire</u>: Le problème du mot est décidable
Soit G une grammaire et $w \in \Sigma^*$ un mot
- On a déjà vu comment décider si $\varepsilon \in \mathcal{L}_G(S)$ en temps $O(|G|)$
- si $w \neq \varepsilon$, calculer G' équivalente propre et réduite. Si $x \xrightarrow{n} |w|$ alors $n < 2|w|$ car chaque dérivation ajoute une lettre terminale ou augmente strictement la longueur du mot $\square$

Forme normale de Chomsky FNC $\mathcal{O}(|G|^2)$

Soit $G = (\Sigma, V, P, S)$

1) Mettre la grammaire en FNQF $\mathcal{O}(|G|)$
2) Supprimer les règles $x \to \varepsilon$ $\mathcal{O}(|G|)$
3) Introduire de nouvelles variables $\mathcal{O}(|G|)$
$(x_a)_{a \in \Sigma}$ et les règles $x_a \to a$
Dans les règles $x \to \alpha_1 \alpha_2$, remplacer les lettres terminales par les variables correspondantes
ex : $x \to ay$ devient $x \to x_a y$
les règles sont maintenant
$$P' \subseteq V' \times (\Sigma \cup V' \cup V'V')$$
4) Supprimer les règles $x \to y$ $\mathcal{O}(|V| \cdot |G|)$

La grammaire est en FNC
on peut éventuellement la réduire.