Simplifications & Formes Normales

Simplifications et formes normales

- L, langage algébrique non vide, engendré par G :
  - sans variable improductive ($X \in N$ est productif s'il existe $w \in T^*$ tq $X \to^* w$)
  - sans variable inaccessible ($X \in N$ est accessible s'il existe $\alpha, \beta \in (N \cup T)^*$: $S \to^* \alpha X \beta$)
- On supprime les variables improductives et les variables inaccessibles
- Si $\varepsilon \notin L$, on n'a pas besoin de production de la forme $A \to \varepsilon$

Simplifications et formes normales

- Toutes les productions peuvent être de la forme $A \to BC|a$ ($A,B,C \in N$ et $a \in T$)
  (Forme Normale de **Chomsky**)
- Toutes les productions peuvent être de la forme $A \to a\alpha$ ($a \in T$ et $\alpha \in N^*$)
  (Forme Normale de **Greibach**)
- Supprimer les règles de renommage (productions de la forme $A \to B$)

Suppression des ε-productions

- Observation: si $\varepsilon \in L(G)$, il sera impossible de supprimer toutes les ε-productions de G; Si $\varepsilon \notin L(G)$, aucune règle de la forme $A \to \varepsilon$ n'est utile.
- Principe:
  - Pour chaque variable A, déterminer si $A \to^* \varepsilon$. Si c'est le cas, on dit que A est «effaçable»
  - Remplacer toute production de la forme $B \to X_1 X_2 \ldots X_n$ par toutes les productions où on a effacé les variables X_i effaçables sans ajouter $B \to \varepsilon$, même si tous les X_i sont effaçables

Variables effaçables

- Une variable A est effaçable si $A \to^* \varepsilon$
- L'ensemble des variables effaçables de G est
  $$Eff(G) = \{X \in N: X \to^* \varepsilon\}$$
  $X \in Eff(G)$

SSI

$X \to \varepsilon \in R$

ou

$X \to X_1 X_2 \ldots X_n \in R$ pour $X_i \in Eff(G)$ $0<i<n+1$

Variables effaçables

Construction inductive:

Base : $Eff_0(G) = \{X: X \to \varepsilon \in R\}$

Induction :

$Eff_i(G) = Eff_{i-1}(G) \cup \{Y: Y \to Z_1 Z_2 \ldots Z_n \in R$ pour $Z_k \in Eff_{i-1}(G)$ $0<k<n+1\}$

On s'arrête dès que $Eff_i(G) = Eff_{i-1}(G)$

Exemple

$S \to aAb|AB|a$

$A \to \varepsilon|AAB$

$B \to AC|b$

$C \to \varepsilon|aba$

Initialisation : variables de la forme $A \to \varepsilon$:
$Eff_0(G)=\{A,C\}$

variables de la forme
$A \to \alpha$, $\alpha \in (Eff_0(G))^*$
$Eff_1(G)=\{A,C\} \cup \{B\}$
variables de la forme
$A \to \alpha$, $\alpha \in (Eff_1(G))^*$
$Eff_2(G)=\{B,A,C\} \cup \{S\}$

Toutes les variables sont effaçables et $\varepsilon \in L(G)$

Supprimer les ε-productions

- On construit un nouvel ensemble de règles R'.
- $X \to X_1X_2...X_n \in R$, $X_i \in (N \cup T)$, remplacée par

$$X \to \alpha_1\alpha_2...\alpha_n$$

Si $X_i \notin Eff(G)$, alors $\alpha_i=X_i$

Si $X_i \in Eff(G)$, alors $\alpha_i=X_i$ et $\alpha_i=\varepsilon$

Les α_i sont non tous nuls

(sauf lorsque toutes les variables sont effacées)

Exemple

- $S \to aAb|AB|a|ab|A|B$
- $A \to \cancel{\varepsilon}|AAB|A|B|AB|AA$
- $B \to AC|b|A|C$
- $C \to \cancel{\varepsilon}|aba$

- Dans G, $S \to \varepsilon$;
  si on veut engendrer le même langage, on ajoute $S \to \varepsilon$

- $Eff(G)=\{B,A,C,S\}$
- En retirant les ε-productions, on obtiendra $L \backslash \{\varepsilon\}$

Conséquence

Théorème : Tout langage algébrique ne contenant pas le mot vide peut être engendré par une grammaire algébrique sans symbole inutile ni ε-production

Il suffit d'appliquer successivement les transformations précédentes : retirer les improductifs; retirer les inaccessibles; retirer les ε-productions

Suppression des renommages

Règles de la forme $X \to Y$ pour $X,Y \in N$

Les variables de renommage

Les variables en lesquelles X peut être renommée sont : $Ren(X)=\{Z \in N : X \to^* Z\}$

Construction inductive, pour chaque variable X, calculer Ren(X)

Base : $Ren_0(X)=\{X\}$

Induction :

$Ren_i(X)=Ren_{i-1}(X) \cup \{Z : Y \to Z \in R \text{ pour } Y \in Ren_{i-1}(X)\}$

On s'arrête dès que $Ren_i(X) = Ren_{i-1}(X)$

Supprimer les règles de renommage : pour chaque $Y \in Ren(X)$ et chaque règle $Y \to \alpha$ qui n'est pas renommage, remplacer $X \to Y$ par $X \to \alpha$

Exemple

$Ren_i(X)=Ren_{i-1}(X) \cup \{Z: Y\to Z\in R,\ Y\in Ren_{i-1}(X)\}$

$S\to ab|aAb|A|B|AB|a$
$A\to A|AA|AB|B|AAB$
$B\to A|C|AC|b$
$C\to aba$

- $Ren_0(S)=\{S\}$
- $Ren_1(S)=\{Z: S\to Z\in R\}\cup\{S\}=\{S,A,B\}$
- $Ren_2(S)=\{Z: Y\to Z\in R, Y\in\{S,A,B\}\} \cup\{S,A,B\} = \{S,A,B,C\}$
- $Ren_0(A)=\{A\}$
- $Ren_1(A)=\{Z: A\to Z\in R\}\cup\{A\} = \{A,B\}$
- $Ren_2(A)=\{Z: Y\to Z\in R, Y\in\{A,B\}\} \cup \{A,B\} = \{A,B,C\}$
- $Ren_0(B)=\{B\}$
- $Ren_1(B)=\{Z: B\to Z\in R\}\cup\{B\} = \{A,B,C\}$
- $Ren_2(B)=\{Z: Y\to Z\in R, Y\in\{A,B,C\}\} \cup\{A,B\} = \{A,B,C\}$
- $Ren_0(C)=\{C\}$
- $Ren_1(C)=\{Z: C\to Z\in R\}\cup\{C\}=\{C\}$

Exemple

- $Ren(S)=\{S,A,B,C\}$
- $Ren(A)=\{A,B,C\}$
- $Ren(B)=\{A,B,C\}$
- $Ren(C)=\{C\}$

$S\to ab|aAb|\cancel{A}|\cancel{B}|AB|a$
$A\to \cancel{A}|AA|AB|\cancel{B}|AAB$
$B\to \cancel{A}|\cancel{C}|AC|b$
$C\to aba$

$\left\{\begin{array}{l} \cdot S\to ab|aAb|AB|a|\ AA|AB|AAB|AC|b|aba \\ \cdot A\to AA|AB|AAB|AC|b|aba \\ \cdot B\to AA|AB|AAB|AC|b|aba \\ \cdot C\to aba \end{array}\right.$

A et B sont identiques

$\left\{\begin{array}{l} \cdot S\to ab|aAb|AA|a|AAA|AC|b|aba \\ \cdot A\to AA|AAA|AC|b|aba \\ \cdot C\to aba \end{array}\right.$

On peut remonter $C\to aba$

$\left\{\begin{array}{l} \cdot S\to ab|aAb|AA|a|AAA|Aaba|b|aba \\ \cdot A\to AA|AAA|Aaba|b|aba \end{array}\right.$

Conséquence

Théorème : A toute grammaire algébrique G ne contenant pas ε correspond une grammaire algébrique équivalente sans règle de renommage

- Il faudrait montrer la double inclusion par récurrence sur la longueur des dérivations

Théorème général

Théorème : Tout langage algébrique ne contenant pas le mot vide peut être engendré par une grammaire sans symbole inutile ni ε-production, ni règle de renommage.

conséquence directe des résultats précédents.

Formes normales

- On peut mettre toutes les productions sous la forme $A\to BC|a$ avec $A,B,C \in N$ et $a\in T$ (FNC)

Forme normale de Chomsky

Forme normale de Chomsky

- **Théorème** : Toute grammaire algébrique sans renommage ni ε-production est équivalente à une grammaire dont les productions sont de la forme
  $X \to a$, pour $a \in T$ et $X \to YZ$, pour $Y, Z \in N$
- Il faut coder un arbre d'arité finie quelconque en un arbre binaire.
- Si une règle est de la forme $X \to \alpha_1\alpha_2\ldots\alpha_n$ on regroupe les variables du membre droit en deux paquets en introduisant des productions supplémentaires

19

Démonstration

Dans chaque production où a est dans le membre droit : $X \to aY$ ou $X \to Ya$

On remplace a par C_a: avec $C_a \to a$: $X \to C_aY$ ou $X \to YC_a$

Les règles sont de la forme (G sans règle de renommage)

$X \to \alpha_1\alpha_2\ldots\alpha_n$ $\quad \alpha_i \in N$

$X \to a$ $\quad a \in T$

20

Démonstration (2)

- On introduit de nouvelles variables pour obtenir à partir de

$X \to \alpha_1 \underbrace{\alpha_2 \ldots \alpha_{n-1}\, \alpha_n}$

$Y_1 \to \alpha_2 \underbrace{\alpha_3 \ldots \alpha_{n-1}\, \alpha_n}$

$Y_2 \to \alpha_2\, \alpha_3 \ldots \alpha_{n-1}\, \alpha_n$

…

$Y_{n-2} \to \alpha_n$

- A présent toutes les règles sont de la forme
  $X \to \alpha\beta \quad \alpha,\beta \in N$ ou $X \to a \quad a \in T$

21

Exemple

- La grammaire ETF
  $E \to E+T|T$
  $T \to T*F|F$
  $F \to (E)|a$

- Ren(E)={E,T,F}
- Ren(T)={T,F}
- Ren(F)={F}

hypothèse de mise sous FNC: pas de règle de renommage

On supprime les règles de renommage

$E \to E+T$	$E \to E+T\mid T*F \mid(E)\mid a$
$T \to T*F$	$T \to T*F \mid(E)\mid a$
$F \to (E)\mid a$	$F \to (E)\mid a$

On peut maintenant mettre cette grammaire sous FNC

22

Exemple

$E \to E+T$	$E \to EX_1$ et $X_1 \to C_+ T$
$E \to T*F$	$E \to TX_2$ et $X_2 \to C_* F$
$E \to (E)$	$E \to C_(X_3$ et $X_3 \to E C_)$
$E \to a$	$E \to a$

$C_(\to ($
$C_) \to)$
$C_* \to *$
$C_+ \to +$

$E \to EX_1|TX_2|C_(X_3|a$
$X_1 \to C_+T$
$X_2 \to C_*F$
$X_3 \to EC_)$

23

Exemple

$T \to T*F$	$T \to TY_1$ et $Y_1 \to C_*F$
$T \to (E)$	$T \to C_(Y_2$ et $Y_2 \to E\, C_)$
$T \to a$	$T \to a$

$T \to TY_1|\, C_(Y_2|a$
$Y_1 \to C_*F$
$Y_2 \to E\, C_)$

$F \to (E)$	$F \to C_(Z_1$ et $Z_1 \to EC_)$
$F \to a$	$F \to a$

24

Exemple

$E \to E+T|T$
$T \to T*F|F$
$F \to (E)|a$

$C_(\to ($ $C_) \to)$ $C_* \to *$ $C_+ \to +$
$E \to EX_1|TX2|C_(X_3|a$
$X_1 \to C_+T$
$X_2 \to C_*F$
$X_3 \to EC_)$
$T \to TY_1|\ C_(Y_2|a$
$Y_1 \to C_*F$
$Y_2 \to E\ C_)$
$F \to C_(Z_1$
$Z_1 \to EC_)$
$F \to a$

25

Suppression de la récursivité gauche

Pour la forme normale de Greibach on
a besoin de la suppression de la récursivité gauche

26

Suppression récursivité gauche

Lemme : Pour toute grammaire algébrique, il existe une grammaire algébrique équivalente sans récursivité gauche i.e. sans règle de la forme

$A \to A\alpha \qquad \alpha \in (N \cup T)^*$

- Transformer la récursivité gauche en récursivité droite.
- On suppose que *G* ne contient que des règles de la forme $A \to A\alpha|\beta$ avec $\alpha,\beta \in (N \cup T)^*$ où β ne commence pas par *A*.
- Ce couple de règles se dérive en $\beta\alpha^*$
- Qu'on peut engendrer par les règles sans récursivité gauche
  - $A \to \beta|\beta X$
  - $X \to \alpha|\alpha X$

27

Plus généralement

- On remplace toute règle de la forme
  $A \to A\alpha_1|\ A\alpha_2|\ldots|A\alpha_n|\beta_1|\ \beta_2|\ldots\ |\ \beta_m$
- Qui engendre une expression de la forme
  $(\beta_1+\beta_2+\ldots+\beta_m)(\alpha_1+\alpha_2+\ldots+\alpha_n)^*$
- Par les règles
  - $A \to \beta_1|\beta_2|\ldots|\beta_m|\beta_1B|\beta_2B|\ldots\ |\beta_mB$
  - $B \to \alpha_1|\alpha_2|\ldots|\alpha_n|\alpha_1B|\alpha_2B|\ldots|\alpha_nB$

28

Exemple

- $E \to EX_1|TX_2|C_(X_3|a$
  $A \to A\alpha_1|A\alpha_2|\ldots|A\alpha_n|\beta_1|\beta_2|\ldots\ |\beta_m$
- est transformée en
  - $E \to TX_2|C_(X_3|a|TX_2B|C_(X_3B|aB$
  - $B \to X_1|X_1B$

$A \to \beta_1|\beta_2|\ldots|\beta_m|\beta_1B|\beta_2B|\ldots\ |\beta_mB$
$B \to \alpha_1|\alpha_2|\ldots|\alpha_n|\alpha_1B|\alpha_2B|\ldots|\alpha_nB$

29

Forme normale de Greibach

30

Théorème

Théorème : Toute grammaire algébrique, sans renommage ni ε-production est équivalente à une grammaire dont toutes les productions sont de la forme

$$X \to a\gamma, \quad \gamma \in (N \cup T)^* \text{ et } a \in T$$

On cherche

• à supprimer la récursivité gauche des règles, i.e. à éviter des règles de la forme $A \to A\alpha$, $\alpha \in (N \cup T)^*$

• à faire commencer toute règle par un terminal.

pour faciliter l'analyse d'une chaîne de caractères

Mise sous FNG (1)

1. On fixe un ordre sur les variables de G: $N=(A_1, A_2, \ldots, A_n)$ et on dit que la grammaire est montante si toute règle est de la forme

   $A_j \to A_k v$ avec $j<k$ ou $A_j \to av$ avec $a \in T$

2. On suppose les variables de G ordonnées et ses règles de la forme

   $A \to A_{i1} A_{i2} \ldots A_{in}$ tq tous les $A_{ij} \in N$ ou $A \to b$ tq $b \in T$

Mise sous FNG (2)

3. On pose i =1. Tant que i < n, on considère les productions P de la forme $A_i \to \alpha$:
   - Si i>1, on remplace toutes les occurrences de A_k , k<i dans les productions $A_i \to A_k\alpha$ avec k<i et $\alpha \in V^*$
   - Soit P est montante OK
   - Soit P récursive gauche
     - on retire la réc. gauche en ajoutant un nouveau non-terminal numéroté à la suite de ceux de N' et on incrémente n de 1
4. pour i de n-1 à 1 on substitue dans les $A_i \to A_j\alpha$ avec j>i les A_j par leurs parties droites.

Exemple

$\cancel{A_1 \to A_2 A_2 | 0}$

$\cancel{A_2 \to A_1 A_2 | 1}$

$\cancel{A_2 \to A_2 A_2 A_2 | 0A_2 | 1}$

$A_2 \to 0A_2 | 1 | 0A_2A_3 | 1A_3$

$\cancel{A_3 \to A_2 A_2 | A_2 A_2 A_3}$

$A_3 \to 0A_2 A_2 \mid 0A_2A_2A_3$
$\mid 1A_2 \mid 1A_2A_3$
$\mid 0A_2A_3 A_2 \mid 0A_2A_3A_2A_3$
$\mid 1A_3A_2 \mid 1A_3A_2A_3$

A_1 montante. On passe à A_2

On copie le membre droit de A_1 à la place de A_1 dans $A_2 \to A_1$ v:

A_2 est récursive gauche, on la transforme grâce au lemme:

On copie le membre droit de A_2 à la place de A_2 dans $A_3 \to A_2$v:

A_3 est sous FNG

A_2 est sous FNG

On copie le membre droit de A_2 à la place de A_2 dans $A_1 \to A_2$ v:

$A_1 \to 0A_2A_2 | 1A_2 | 0A_2A_2A_2 | 1ZA_2 | 0$ FNG