

Analyse descendante

Etant données :

- Une grammaire G
- Une suite d'unités lexicales ω

on cherche à construire une dérivation gauche pour ω

Ou encore

on cherche à construire un arbre d'analyse pour ω en partant de S (le start) et en créant les nœuds de l'arbre en préordre (RGD)

le symbole courant (Look ahead) permet de guider le choix de la bonne production

63

Analyse prédictive

Étant donnée une configuration :

protophrase texte
 $\dots X \dots$ $\dots a \dots$

on veut pouvoir déterminer « à coup sûr » quelle règle $X \rightarrow \alpha$ doit être choisie

Exemple :

	protophrase	texte	règle choisie
$S \rightarrow ABB \mid BAA$	$\dots A \dots$	$\dots a \dots$	$A \rightarrow a$
$A \rightarrow a \mid b$	$\dots A \dots$	$\dots b \dots$	$A \rightarrow b$
$B \rightarrow c$	$\dots c \dots$	$\dots c \dots$	aucune
	$\dots B \dots$	$\dots a \dots$	aucune (idem b)
	$\dots S \dots$	$\dots c \dots$	$B \rightarrow c$
	$\dots S \dots$	$\dots a \dots$	$S \rightarrow ABB$
	$\dots S \dots$	$\dots b \dots$	$S \rightarrow ABB$
	$\dots S \dots$	$\dots c \dots$	$S \rightarrow BAA$

64

Analyse prédictive

Méthode d'analyse descendante sans back-track où un seul symbole de Look Ahead (LA) suffit à déterminer le choix de la bonne production lors du développement d'un NT X

- Pour certaines grammaires, dites $LL(1)$, on peut construire automatiquement des analyseurs prédictifs.

Ces grammaires doivent être :

- non ambiguës
- sans récursivité gauche
- factorisées à gauche

(conditions nécessaires mais non suffisantes)

65

Si une grammaire est ambiguë

Il existe une phrase ω du langage qui admet 2 arbres d'analyse (ou 2 dérivations gauches)

Exemple :

$A \rightarrow aB \mid B$
 $B \rightarrow b \mid ab$

$\omega = ab$

1. $A \Rightarrow_g aB \Rightarrow_g ab$

2. $A \Rightarrow_g B \Rightarrow_g ab$

Quand on doit développer A , on se retrouve forcément devant un choix non déterministe ($A \rightarrow aB$ ou $A \rightarrow B$) puisque les 2 règles « marchent »

\Rightarrow on ne peut pas choisir « à coup sûr » une règle unique

66

Si une grammaire est récursive à gauche

Exemple :

$S \rightarrow Sa \mid b$ (génère les mots de la forme ba^*)

protophrase	texte	règle choisie
$\dots S \dots$	$\dots a \dots$	aucune
$\dots S \dots$	$\dots b \dots$?

On se retrouve devant un choix impossible ($S \rightarrow Sa$ ou $S \rightarrow b$) puisqu'aucune des 2 règles ne marche « à coup sûr » (i.e., dans tous les cas)

67

Si une grammaire a des facteurs gauches communs

Exemple :

$S \rightarrow aaS \mid abS \mid c$

protophrase	texte	règle choisie
$\dots S \dots$	$\dots a \dots$?

Il faudrait 2 symboles de prévision pour pouvoir choisir entre les 2 règles ($S \rightarrow aaS$ ou $S \rightarrow abS$)

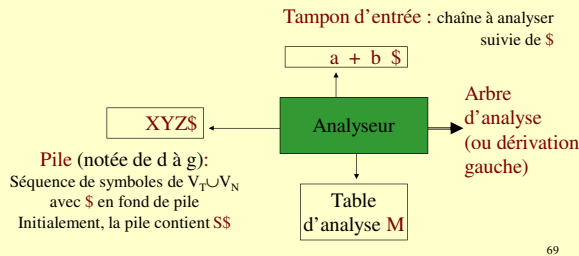
\Rightarrow aucune des 2 règles ne marche « à coup sûr » avec un seul symbole de LA

Grammaire factorisée :	protophrase	texte	règle choisie
$S \rightarrow aS' \mid c$	$\dots S \dots$	$\dots a \dots$	$S \rightarrow aS'$
$S' \rightarrow aS \mid bS$	$\dots S \dots$	$\dots b \dots$	aucune
	$\dots S \dots$	$\dots c \dots$	$S \rightarrow c$
	$\dots S' \dots$	$\dots a \dots$	$S' \rightarrow aS$
	$\dots S' \dots$	$\dots b \dots$	$S' \rightarrow bS$
	$\dots S' \dots$	$\dots c \dots$	aucune

68

Analyseur itératif

- On utilise une table d'analyse qui détermine la règle à appliquer en fonction du NT X à développer et du prochain symbole a du texte (ex. au tableau)
- Modèle d'analyseur prédictif



69

Fonctionnement de l'analyseur

Soit X le sommet de pile, et a le symb. d'entrée courant (LA)

- Si $X = \$$ et $a = \$$ alors succès, FIN
- Si $X = a$ ($\neq \$$) alors - dépiler a
- lire le symbole d'entrée suivant

- Si $X \in V_N$ alors
si $M[X, a] = \{X \rightarrow X_1 X_2 \dots X_n\}$ alors
- dépiler X
- empiler X_1, X_2, \dots, X_n (de d à g)
(- imprimer $X \rightarrow X_1 X_2 \dots X_n$)

- Sinon erreur (appel de proc. de récupération sur erreur)
Répéter cette procédure jusqu'à ce que $X = \$$ (pile vide)

70

Construction de la table d'analyse

Problème : étant donnée une règle $X \rightarrow \alpha$, on veut déterminer par quels symboles peut commencer α (pour savoir dans quelle case de la table M on va mettre la règle)

On utilise deux fonctions : **Premier** et **Suivant**

- Déf : soit $\alpha \in (V_T \cup V_N)^*$
Premier(α) est l'ensemble des terminaux qui débutent les chaînes qui se dérivent de α

i.e., $\text{premier}(\alpha) = \{a \in V_T / \alpha \Rightarrow^* a w\}$

71

Calcul de Premier(α)

- $\text{premier}(a) = \{a\}$ si $a \in V_T$
- $\text{premier}(A) = \text{premier}(\alpha_1) \cup \dots \cup \text{premier}(\alpha_p)$
si $A \rightarrow \alpha_1 \mid \dots \mid \alpha_p$
sont toutes les A-productions ($A \in V_N$)
- $\text{premier}(X\beta) = \text{premier}(X)$ si X n'est pas effaçable
 $= \text{premier}(X) \cup \text{premier}(\beta)$ si X est effaçable
($X \in V_T \cup V_N$)

72

Rappel du problème : étant donnée une règle $X \rightarrow \alpha$, on veut déterminer par quels symboles peut commencer α pour savoir dans quels cas on choisira la règle

- Cas 1 :** il n'y a pas d' ϵ -productions
La règle $X \rightarrow \alpha$ sera choisie si :
- le symbole NT à développer est X
- le symbole courant du texte a peut débuter α : $a \in \text{premier}(\alpha)$
- Cas 2 :** il y a des ϵ -productions
La règle $X \rightarrow \epsilon$ sera choisie si :
- le symbole NT à développer est X
- le symbole courant du texte a peut suivre X : $a \in \text{suivant}(X)$

73

suivant(X)

- Déf : soit $X \in V_N$
Suivant(X) est l'ensemble des terminaux qui peuvent apparaître immédiatement à droite de X dans une protophrase

(plus $\{ \$ \}$ si X peut être le dernier symbole d'une protophrase)
i.e., $\text{suivant}(X) = \{a \in V_T / S \Rightarrow^* \alpha X a \beta, \alpha, \beta \in (V_T \cup V_N)^*\} \cup \{ \$ \}$ si $S \Rightarrow^* \alpha X$

- Calcul de **suivant**(X) :
(1) $\$ \in \text{suivant}(S)$
(2) pour chaque règle $A \rightarrow \alpha X \beta$
- Ajouter **premier**(β) à **suivant**(X)
- Si β est effaçable (ou $\beta = \epsilon$) alors ajouter **suivant**(A) à **suivant**(X)

74

Construction de la table M

Algo : soit une grammaire G

- Pour chaque règle $X \rightarrow \alpha$ faire
 - Pour chaque terminal $a \in \text{premier}(\alpha)$
mettre $X \rightarrow \alpha$ dans $M[X,a]$
 - Si α est effaçable (ou $\alpha = \epsilon$) alors,
pour chaque $b \in \text{suivant}(X)$,
mettre $X \rightarrow \alpha$ dans $M[X,b]$
- Les entrées non définies sont des erreurs

75

exercice

$E \rightarrow E+F \mid F$
 $F \rightarrow F*G \mid G$
 $G \rightarrow \text{id}$

1. Préparer la grammaire :
 - supprimer d'éventuelles ambiguïtés
 - supprimer la récursivité à gauche
 - factoriser (à gauche)
2. Calculer **Premier** et **Suivant**
3. Construire la table d'analyse
4. Faire fonctionner l'analyseur sur la phrase $\text{id}+\text{id}*\text{id}$ et dessiner l'arbre d'analyse construit en explicitant l'ordre de sa construction

76

Grammaires LL(1)

- Une grammaire est **LL(1)** si chaque case de la table M contient au plus une règle
- LL(1) signifie :
 - Left : parcours de l'entrée w de gauche à droite
 - Left : construction d'une dérivation gauche
 - 1 : un seul symbole de prévision (LA)
- On peut montrer que l'algo produit, pour toute grammaire LL(1) G, une table qui analyse toutes les phrases de L(G), et seulement celles-là

77

exercice

Pour chacune des grammaires suivantes, calculer les fonctions premier et suivant, et construire la table d'analyse

Essayer de comprendre ce qui empêche chacune de ces grammaires d'être LL(1)

1. $S \rightarrow Sb \mid a$
2. $S \rightarrow AB \mid BA$
 $A \rightarrow a \mid b$
 $B \rightarrow c \mid \epsilon$
3. $S \rightarrow AB$
 $A \rightarrow c \mid \epsilon$
 $B \rightarrow c$
4. $S \rightarrow aSa \mid \epsilon$

79

Récapitulatif

1. Préparer la grammaire :
 - supprimer d'éventuelles ambiguïtés
 - supprimer la récursivité à gauche
 - factoriser (à gauche)
 2. Calculer **Premier** et **Suivant**
 3. Construire la table d'analyse
- S'il n'y a pas plus d'une règle par case
alors c'est bon, on a construit un analyseur
sinon il faut changer de grammaire
ou changer de méthode d'analyse

80