3 - Grammaires algébriques et analyse syntaxique

1. Grammaires syntagmatiques et grammaires algébriques
2. Définitions
3. Pouvoir d'expression et hiérarchie de Chomsky
4. Analyse syntaxique ascendante tabulaire

3.1 - Grammaires syntagmatiques et grammaires algébriques

- Les grammaires syntagmatiques sont fondées sur l'idée que les mots d'une phrase se regroupent en **syntagmes** qui constituent des entités autonomes.

- Pour une langue donnée, la délimitation des syntagmes dans les énoncés et leur classement par catégories sont fondés sur des critères **distributionnels**.

- Presque simultanément, Chomsky (1956) et Backus (1959) ont conçu un formalisme, les **grammaires algébriques**, qui a été utilisé pour représenter les grammaires syntagmatiques.

3.1 - Grammaires syntagmatiques et grammaires algébriques

- Critère d'interchangeabilité :

  Jean vient.

  Le printemps vient.

  Le garçon vient.

  Un garçon vient.

  Jean voit un garçon.

  Jean connaît un garçon.

- Critère d'antéposition et de postposition :

  C'est un garçon que Jean connaît.

  **C'est un que Jean connaît garçon.*

3.2 - Définitions

- Une grammaire algébrique est un quadruplet (N, T, S, R) tel que:
  - ✓ N est un alphabet fini de symboles non terminaux.
  - ✓ T est un alphabet fini de symboles terminaux.
  - ✓ S est un élément particulier de N, le symbole de départ.
  - ✓ R est un ensemble fini de règles de production de la forme $A \rightarrow \alpha$, où A est un non terminal et α un mot de $(N \cup T)^*$.
- La relation de **dérivation** $\Rightarrow$ entre mots de $(N \cup T)^*$ est définie ainsi : $\alpha_1 A \alpha_2 \Rightarrow \alpha_1 \alpha \alpha_2$ si $A \rightarrow \alpha \in R$. Sa clôture réflexive et transitive est notée : $\Rightarrow^*$
- Le langage engendré par la grammaire est l'ensemble des mots α de T^* tels que : $S \Rightarrow^* \alpha$. Un langage engendré par une grammaire algébrique est appelé un **langage algébrique**.

3.2 - Définitions

- Grammaire algébrique définie par les règles ci-dessous et le symbole de départ S:

S → NP Vintr

S → NP Vtr NP

S → NP Vc Compl

S → S PP

NP → Det N

NP → NP PP

Compl → Conj S

PP → Prep NP

NP → jean

Det → le

N → bébé

N → berceau

Vintr → dort

Vtr → porte

Vc → pense

Prep → dans

Conj → que

3.2 - Définitions

- Une **dérivation** d'un mot α du langage est une suite $S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow \alpha$.

  Elle est représentée de manière compacte par un **arbre d'analyse**. Un arbre d'analyse est un arbre ordonné étiqueté par des symboles de $N \cup T$. La racine est étiquetée par S. Les feuilles sont étiquetés par des symboles terminaux. Chaque nœud qui n'est pas une feuille est étiqueté par un non terminal A et ses fils ordonnés sont étiquetés par des symboles constituant un mot α tel que $A \rightarrow \alpha$ est une règle de production de la grammaire.

- Deux grammaires sont faiblement **équivalentes** si elles engendrent le même langage.

- Une grammaire est **ambiguë** si elle permet de produire deux arbres d'analyse différents pour le même mot.

3.2 - Définitions : exercices

1. De la grammaire au langage

Déterminer les langages engendrés par les grammaires suivantes (dans chacun des cas, S est le symbole de départ). S'ils sont ambigus, le montrer à l'aide d'un exemple.

a) $S \rightarrow \varepsilon \mid aaaS$

b) $S \rightarrow SS \mid a$

c) $S \rightarrow aA \mid Bb \qquad A \rightarrow Ab \mid b \qquad B \rightarrow aB \mid a$

d) $S \rightarrow \varepsilon \mid aSa \mid bSb \mid cSc$

2. Du langage à la grammaire

Définir des grammaires algébriques engendrant les langages suivants.

a) $L_1 = \{ a^n b^p \mid 0 < p < n \}$

b) $L_2 = \{ a^n b^n c^m d^m \mid n, m \in \mathbb{N} \}$

c) $L_3 = \{ a^n b^m c^p \mid n = m \text{ or } p = m \}$

3.2 - Définitions : exercices

3. **Grammaire du français**

Etendre la grammaire du français du cours et avec cette grammaire, analyser les phrases suivantes :

a) *Le beau bébé dans le berceau dort profondément.*

b) *Jean voit le bébé endormi dans le berceau dans la chambre.*

c) *Jean pense que dans le berceau le bébé dort profondément.*

d) *Que voit Jean ?*

e) *Que voit Jean dans la chambre ?*

f) *Le beau bébé que Jean voit dort.*

g) *Le beau bébé que Jean pense que Marie garde dort dans la chambre.*

Discuter de la pertinence de la grammaire construite par rapport au problème de la surgénération

3.3 - Pouvoir d'expression et hiérarchie de Chomsky

- Les grammaires algébriques permettent d'exprimer la **récursivité** des langues naturelles.

- Une grammaire algébrique est **récursive** si elle produit une dérivation de la forme $A \Rightarrow^{+} \alpha_1 A \alpha_2$

- Les grammaires algébriques récursives, sous certaines conditions de bonne formation, sont celles qui engendrent les langages algébriques **infinis**.

- Les langages réguliers sont des langages algébriques. Ce sont les langages engendrés par les **grammaires régulières** gauches ou droites. Une grammaire régulière gauche (droite) est une grammaire algébrique où toutes les règles ont la forme : $A \rightarrow B\ \alpha$ $(A \rightarrow \alpha\ B)$

3.3 - Pouvoir d'expression et hiérarchie de Chomsky

- En relâchant la forme des règles des grammaires algébriques, nous obtenons des grammaires avec un pouvoir d'expression plus élevé.
- Il est possible de construire une hiérarchie de 4 classes numérotées de 0 à 3 : la **hiérarchie de Chomsky**.
- Chaque classe de la hiérarchie contient strictement les classes avec un numéro plus élevé : le pouvoir d'expression diminue lorsque le numéro augmente.
- Dans le même temps, la complexité des machines dédiées à la reconnaissance des langages pour chaque classe diminue lorsque le numéro augmente.

3.3 - Pouvoir d'expression et hiérarchie de Chomsky

Type	Nom	Forme des règles	Complexité de la reconnaissance
0	Turing équivalent	$\alpha \rightarrow \gamma$ tel que : $\alpha \neq \varepsilon$	= Langages énumérables récursivement par une machine de Turing
1	Sensible au contexte	$\alpha A \beta \rightarrow \alpha \gamma \beta$	$\subset$ Languages récursifs reconnus par une machine de Turing
2	Non contextuel ou algébrique	$A \rightarrow \gamma$	= Langages reconnus par un automate à pile
3	régulier	$A \rightarrow \gamma B$ ou $A \rightarrow \gamma$ avec $\gamma \in T^*$	= Langages reconnus par un automate d'états finis

3.3 - Pouvoir d'expression et hiérarchie de Chomsky

- Etant donné la spécificité des langues naturelles (limites floues et en constante évolution, interactions entre les différents niveaux), elles ne peuvent être qu'approchées par des langages formels.

- La classe des langages algébriques n'est pas suffisamment expressive pour représenter certaines langues (la syntaxe du suisse allemand, la morphologie du bambara …)

- La démonstration fait appel à la distinction entre **compétence** et **performance** au niveau du cerveau humain. La compétence est liée à la grammaire du langage alors que la performance touche aux ressources humaines utilisées pour analyser et engendrer des énoncés.

- L'adéquation des formalismes linguistiques ne doit pas seulement être évaluée à l'aune des langages engendrés mais aussi en examinant leur faculté d'exprimer les généralités linguistiques (par exemple les alternances).

3.4 - Analyse syntaxique ascendante tabulaire

- Le but de l'analyse syntaxique est de trouver tous les arbres d'analyse pour une phrase et une grammaire données.

- Une **analyse ascendante** ou guidée par les données cherche à construire les arbres d'analyse des feuilles à la racine. Les règles de la grammaire sont utilisées de droite à gauche.

- Une **analyse descendante** ou guidée par le but cherche à construire les arbres d'analyse de la racine vers les feuilles. Les règles de la grammaire sont utilisées de gauche à droite.

- L'inconvénient d'une analyse ascendante est de construire des arbres inutiles qui n'entreront jamais dans un arbre dont la racine représente une phrase. L'inconvénient d'une analyse descendante est de construire d'autres arbres inutiles dont la liste des feuilles entre en contradiction avec la phrase à analyser.

3.4 - Analyse syntaxique ascendante tabulaire

- La **tabulation** ou **mémoïsation** est une technique caractéristique de la **programmation dynamique** qui utilise une table pour stocker les résultats intermédiaires pour éviter de les recalculer en cas de besoin.
- Les données calculées sont rangées sous forme d'items dans une **table**.
- Dans le cas de l'analyse syntaxique, les règles de calcul des items peuvent être présentées sous forme d'un **système déductif**.
- La signification des items est donnée par un **invariant** qui est préservé par les règles de déduction.
- Pour gérer la façon dont les règles sont appliquées, on utilise un **agenda** où sont stockés les items actifs, déclencheurs des règles et une stratégie d'ordonnancement des items dans l'agenda.

3.4 - Analyse syntaxique ascendante tabulaire

- L'algorithme de Cocke (1970) - Kasami (1965) - Younger (1967) (CKY ou CYK) est une méthode tabulaire d'analyse ascendante des grammaires algébriques.

- Soit $w_1\, w_2 \ldots w_n$ une phrase de n mots à analyser à l'aide d'une grammaire algébrique G.

  L'algorithme de CKY peut être traduit par un système déductif manipulant des items de la forme [i, j, A] avec l'invariant suivant : le segment $w_{i+1} \ldots w_j$ est un mot A (terminal) ou un syntagme de type A (non terminal).

  Les règles d'inférence des items sont les suivantes :

  ---------- Init

  $[\, i, i+1, w_{i+1}]$

  $[i_0, i_1, A_1]\ [i_1, i_2, A_2] \ldots. [i_{p-1}, i_p, A_p]$

  ---------- Compl si $A \rightarrow A_1\, A_2 \ldots. A_p$ est une règle de G

  $[i_0, i_n, A]$

- La complexité en temps de l'algorithme est de l'ordre de n^3

3.4 - Analyse syntaxique ascendante tabulaire

- Grammaire algébrique définie par les règles ci-dessous et le symbole de départ S:

S → NP Vintr

S → NP Vtr NP

S → NP Vc Compl

S → S PP

NP → Det N

NP → NP PP

Compl → Conj S

PP → Prep NP

NP → jean

Det → le

N → bébé

N → berceau

Vintr → dort

Vtr → porte

Vc → pense

Prep → dans

Conj → que

3.4 - Analyse syntaxique ascendante tabulaire : exercices

1. A l'aide de l'algorithme de CKY et de la grammaire construite précédemment en exercice, analyser les phrases :
   a) *Le beau bébé dans le berceau dort profondément.*
   b) *Jean voit le bébé endormi dans le berceau dans la chambre.*
   c) *Jean pense que dans le berceau le bébé dort profondément.*

2. Analyser la phrase « *le joueur de football américain arrive* » en utilisant l'algorithme de CKY avec la grammaire suivante :

S → NP VP	Adj → américain
NP → Det N	Det → le
N → N PP | N Adj	N → football | joueur
PP → Prep N	Prep → de
	VP → arrive